

IMPROVED METHOD AND DEVICE FOR DYNAMIC SHIFT BETWEEN ROUTING AND SWITCHING BURSTS IN DATA TRANSMISSION NETWORK

Publication number: RU2189072 (C2)

Publication date: 2002-09-10

Inventor(s): LAJON TOMAS [US]; N JUMEHN PITER [US]; MINSHOLL GREG [US]; KHINDEN ROBERT [US]; LIAV FONG CHING [US]; KHOFFMAN EHRIK [US]; KH JUSTON LOURENS B [US]; ROBERSON UILL JAM A [US]

Applicant(s): IPSILON NETUORKS INK [US]

Classification:


- international: **G06F13/14; H04L12/46; H04L12/56; H04L12/66; H04L29/06; H04Q11/04; G06F13/14; H04L12/46; H04L12/56; H04L12/66; H04L29/06; H04Q11/04; (IPC1-7): G06F13/14; H04L12/56; H04L12/66**


- European: **H04L12/56S8F; H04L12/46B1; H04L29/06; H04L29/06J; H04Q11/04S2**


Application number: RU19980116258 19970130


Priority number(s): US19960597520 19960131; US19960024272P 19961122


Also published as:

 US5920705 (A)

 WO9728505 (A1)

 JP2002516042 (T)

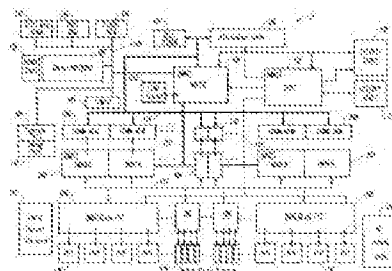
 EP1012730 (A1)

 CN1213444 (A)

[more >>](#)

Abstract of RU 2189072 (C2)

network communications. SUBSTANCE: base switching unit has hardware switching facility and controller; the latter has processor and memory. Gateway switching unit has base switching unit and gateway unit controller that has processor, memory, and plurality of network interface boards. Actuating switching device has base switching unit incorporating switching controller and processor, memory, and plurality of network interface boards. Method describes operation of these devices. EFFECT: enhanced speed and information capacity of network. 50 cl, 21 dwg, 2 tbl



.....
Data supplied from the **esp@cenet** database — Worldwide



(19) **RU** (11) **2 189 072** (13) **C2**
(51) МПК⁷ **G 06 F 13/14, H 04 L 12/56, 12/66**

РОССИЙСКОЕ АГЕНТСТВО
ПО ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

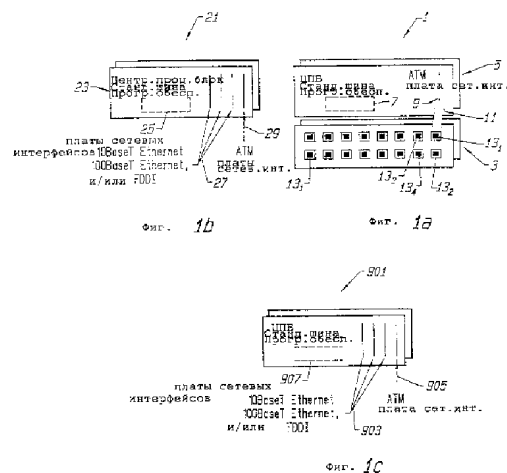
(12) **ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ РОССИЙСКОЙ ФЕДЕРАЦИИ**

(21), (22) Заявка: 98116258/09, 30.01.1997
(24) Дата начала действия патента: 30.01.1997
(30) Приоритет: 31.01.1996 US 08/597,520
22.11.1996 US 60/024,272
(43) Дата публикации заявки: 20.06.2000
(46) Дата публикации: 10.09.2002
(56) Ссылки: US 5379297 A, 03.01.1995. RU 2035065
Cl, 10.05.1995. US 5444702 A, 22.08.1995. EP
0473066 A1, 04.03.1992. EP 051144 A1,
28.10.1992. SU 1695329 A1, 30.11.1991.
(85) Дата перевода заявки РСТ на национальную
фазу: 31.08.1998
(86) Заявка РСТ:
US 97/01595 (30.01.1997)
(87) Публикация РСТ:
WO 97/28505 (07.08.1997)
(98) Адрес для переписки:
129010, Москва, ул. Б. Спасская, 25, стр.3,
ООО "Юридическая фирма Городиский и
Партнеры", Ю.Д.Кузнецову, рег. № 595

(71) Заявитель:
ИПСИЛОН НЕТУОРКС, ИНК. (US)
(72) Изобретатель: ЛАЙОН Томас (US),
НЬЮМЭН Питер (US), МИНШОЛЛ Грег
(US), ХИНДЕН Роберт (US), ЛИАВ Фонг Чинг
(US), ХОФФМАН Эрик (US), ХЬЮСТОН Лоуренс
Б. (US), РОБЕРСОН Уилльям А. (US)
(73) Патентообладатель:
ИПСИЛОН НЕТУОРКС, ИНК. (US)
(74) Патентный поверенный:
Кузнецов Юрий Дмитриевич

(54) **УСОВЕРШЕНСТВОВАННЫЙ СПОСОБ И УСТРОЙСТВО ДЛЯ ДИНАМИЧЕСКОГО СМЕЩЕНИЯ МЕЖДУ ПАКЕТАМИ МАРШРУТИЗАЦИИ И КОММУТАЦИИ В СЕТИ ПЕРЕДАЧИ ДАННЫХ**

(57) Изобретение относится к области сетевых коммуникаций. Техническим результатом является высокое быстродействие и большая информационная емкость сети. Базовый коммутационный блок содержит аппаратное средство коммутации и контроллер, содержащий процессор и память. Шлюзовый блок коммутации содержит базовый коммутационный блок и контроллер шлюзового блока, содержащий процессор, память и множество плат сетевых интерфейсов. Исполнительное устройство коммутации содержит базовый коммутационный блок, содержащий контроллер и процессор коммутации, процессор, память и множество плат сетевых интерфейсов. Способы описывают работу указанных устройств. 11 с. и 39 з.п.ф.-лы, 21 ил., 2 табл.





(19) **RU** ⁽¹¹⁾ **2 189 072** ⁽¹³⁾ **C2**
(51) Int. Cl.⁷ **G 06 F 13/14, H 04 L 12/56, 12/66**

RUSSIAN AGENCY
FOR PATENTS AND TRADEMARKS

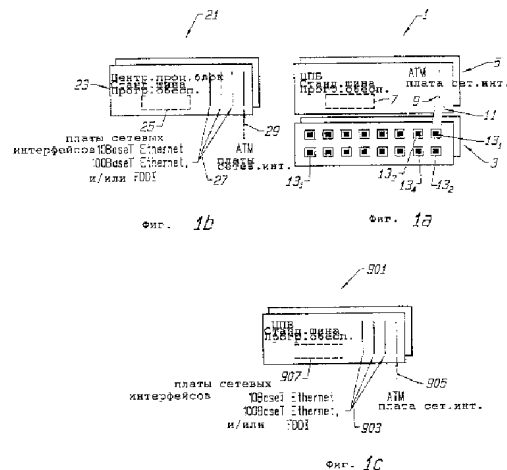
(12) **ABSTRACT OF INVENTION**

(21), (22) Application: 98116258/09, 30.01.1997
(24) Effective date for property rights: 30.01.1997
(30) Priority: 31.01.1996 US 08/597,520
22.11.1996 US 60/024,272
(43) Application published: 20.06.2000
(46) Date of publication: 10.09.2002
(85) Commencement of national phase: 31.08.1998
(86) PCT application:
US 97/01595 (30.01.1997)
(87) PCT publication:
WO 97/28505 (07.08.1997)
(98) Mail address:
129010, Moskva, ul. B. Spasskaja, 25, str.3,
OOO "Juridicheskaja firma Gorodisskij i
Partnery", Ju.D.Kuznetsovu, reg. № 595

(71) Applicant:
IPSILON NETUORKS, INK. (US)
(72) Inventor: LAJON Tomas (US),
N'JuMEhN Piter (US), MINShOLL Greg
(US), KhINDEN Robert (US), LIAV Fong Ching
(US), KhOFFMAN Ehrik (US), Kh'JuSTON
Lourens B. (US), ROBERSON Uill'jam A. (US)
(73) Proprietor:
IPSILON NETUORKS, INK. (US)
(74) Representative:
Kuznetsov Jurij Dmitrievich

(54) **IMPROVED METHOD AND DEVICE FOR DYNAMIC SHIFT BETWEEN ROUTING AND SWITCHING BURSTS IN DATA TRANSMISSION NETWORK**

(57) Abstract:
FIELD: network communications.
SUBSTANCE: base switching unit has hardware switching facility and controller; the latter has processor and memory. Gateway switching unit has base switching unit and gateway unit controller that has processor, memory, and plurality of network interface boards. Actuating switching device has base switching unit incorporating switching controller and processor, memory, and plurality of network interface boards. Method describes operation of these devices. EFFECT: enhanced speed and information capacity of network. 50 cl, 21 dwg, 2 tbl



Предпосылки изобретения

Изобретение относится к области сетевых коммуникаций. Более конкретно в одном из вариантов осуществления настоящее изобретение обеспечивает способ и устройство для динамического смещения между пакетами коммутации и маршрутизации эффективным образом, чтобы обеспечить высокую пропускную способность для передачи пакетов при поддержании полных функциональных возможностей маршрутизации в соответствии с протоколом Интернет (IP). Настоящее изобретение объединяет высокое быстродействие, информационную емкость, возможность многозадачного трафика с простотой, масштабируемостью и надежностью.

Ввиду широкого распространения и непрерывного роста сети Интернет, которая использует протокол IP, указанный протокол превратился в преобладающий протокол сетевого уровня, используемый в настоящее время.

Протокол IP определяет формат блока данных протокола (PDU) и взаимодействие станции с маршрутизатором и маршрутизатора с маршрутизатором. Протокол IP обеспечивает не требующее соединений обслуживание по передаче данных к пользователям протокола IP на станциях, связанных с сетями Интернет. Не требующая соединений модель, на которой основывается протокол IP, обеспечивает надежную и гибкую основу для построения сети комплексных услуг. Все основные операционные системы включают реализацию протокола IP, позволяющую протоколу IP и соответствующему протоколу уровня передачи (уровень 4 или опорная модель OSI (взаимные соединения открытых систем)) т.е. протоколу управления передачей (TCP) использоваться универсально практически на всех платформах аппаратных средств. Одним из основных преимуществ протокола IP является его чрезвычайно высокая масштабируемость, успешно реализуемая в сетях как с небольшим числом пользователей, так и в сетях с размерами, соответствующими предприятиям, и включая глобальную сеть Интернет.

С быстрым ростом сети Интернет обычные маршрутизаторы протокола IP становятся неадекватными по их способности обрабатывать трафик сети Интернет. При современных требованиях к быстродействию рабочих станций сети вычислениям на пользовательском сервере и высоким значениям ширины полосы сети все чаще сталкиваются с проблемами перегруженности трафика. Типовые проблемы включают, например, изменяемые в весьма высокой степени моменты времени отклика сети, высокие значения частоты отказов сети, неспособность поддерживать задачи, чувствительные к задержкам.

Коммутаторы локальных сетей предоставляют относительно экономичный и быстродействующий способ решения проблемы перегрузки в сегментах локальных сетей, совместно использующих носители информации. Технология коммутации представляется более эффективным средством в управлении трафиком и распределении ширины полосы в пределах локальной сети, чем концентраторы

совместно используемых носителей данных или простые мосты для сетевых соединений. Коммутаторы локальных сетей работают как аппаратные средства, пересылающие пакеты на уровне данных (уровень 2 модели OSI), осуществляя обработку адресов управления доступом к носителю и выполняя простые функции табличной перекодировки. Сети, основанные на использовании коммутаторов, способны обеспечивать более высокую пропускную способность, однако им по-прежнему свойственны проблемы, связанные с лавинной маршрутизацией передач и неудовлетворительной защищенностью. Маршрутизаторы, которые работают на сетевом уровне (уровень 3 опорной модели OSI), по-прежнему необходимы для решения подобных проблем. Однако средства быстродействующей коммутации превосходят способности современных маршрутизаторов, создавая "узкие места". Традиционные средства пересылки пакетов протокола IP, на которых базируется сеть Интернет, такие как IP-маршрутизатор, проявляют свою неадекватность. Маршрутизаторы являются дорогостоящими, сложными и характеризуются ограниченной пропускной способностью по сравнению с новыми средствами коммутации. Для поддержки возрастающих потребностей трафика в больших сетях и в Интернет IP-маршрутизаторы должны работать более быстро и с меньшими затратами.

Кроме того, необходим выбор качества обслуживания для поддержки возрастающих потребностей в мультимедийных задачах и в задачах реального времени, включая, например, конференц-связь. Современный протокол управления передачей/протокол Интернет (TCP/IP) не обеспечивает поддержки выбора качества обслуживания. Однако по мере того как прогрессивные функциональные возможности, требуемые все большим числом типов трафика, вводятся в протокол IP, традиционные маршрутизаторы протокола IP все в большей степени оказываются недостаточными для применения в качестве средств пакетной передачи.

Режим асинхронной передачи (ATM) характеризует собой высокоскоростную, масштабируемую, многозадачную технологию, рекламируемую в настоящее время в качестве краеугольного камня современных сетей, не использующих маршрутизаторы. ATM представляет собой высокоэффективную технологию пакетной передачи, настолько отличную от современных сетевых архитектур, таких как IP, что в настоящее время отсутствует простой путь внедрения данной технологии. ATM испытывает затруднения в эффективной поддержке существующего локального сетевого трафика ввиду его ориентированной на соединения архитектуры, что создает необходимость в дополнительной установке очень сложных непроверенных многоуровневых протоколов. Такие протоколы создают очевидные проблемы неприемлемо длительного времени установки соединения коммутируемого виртуального канала. Кроме того, обеспечение возможности пользователям TCP/IP передавать и принимать ATM трафик с использованием

коммутируемых виртуальных каналов требует принятия более новых, непроверенных и чрезвычайно сложных протоколов. Эти протоколы не позволяют в прикладных задачах, реализуемых с использованием протоколов TCP/IP, получать выгоду за счет процедуры выбора качества обслуживания для режима ATM, что влечет за собой очень большой объем непроизводительных расходов для сетевых администраторов, не позволяя использовать ключевые преимущества ATM. Кроме того, многие из этих протоколов дублируют функциональные возможности хорошо установленного комплекта протоколов TCP/IP, а необходимость изучения этих сложных протоколов увеличивает затраты для владельцев средств ATM для сетевых администраторов, которые должны решать проблемы неисправностей в сети. Затруднения в продвижении ATM особенно очевидны в свете проверенного временем и отлаженного протокола IP, надежно поддерживаемого огромной и все возрастающей в размерах пользовательской базы, как это вытекает из популярности сети Интернет.

С учетом неадекватности современных решений проблем поставщика обслуживания разработали новые сетевые архитектуры распределенной маршрутизации. Однако эти архитектуры зачастую являются сложными, запутанными, дублируют функциональные возможности, обеспечиваемые протоколом IP. Такие архитектуры также приводят к увеличению сложности проблем для сетевых администраторов. Например, дублирование функциональных возможностей приводит к возрастанию нагрузки на функцию сетевого управления и может сделать выделение сетевых задач весьма затруднительным. Очевидно, что необходима система, позволяющая избежать "узких мест" и увеличения сложности сетевого управления. Кроме того, необходимо создать сетевую архитектуру, которая совместима с протоколом IP и исключает излишнее дублирование.

Сущность изобретения

Настоящее изобретение относится к области сетевых коммуникаций, в частности к способу и устройству для динамического смещения между пакетами коммутации и маршрутизации эффективным образом, позволяющим обеспечить высокую пропускную способность при передаче пакетов в целях проблем, охарактеризованных выше.

В соответствии с одним из вариантов осуществления настоящее изобретение обеспечивает способ передачи пакетов между узлом восходящего потока и узлом нисходящего потока в сети, причем узел нисходящего потока находится в нисходящем потоке относительно узла восходящего потока. Способ предусматривает этапы установления по умолчанию виртуальных каналов между узлом восходящего потока и узлом нисходящего потока, приема пакета в узле нисходящего потока и выполнения классификации потока в узле нисходящего потока по пакету, чтобы определить, принадлежит ли пакет к специфическому потоку, который должен быть перенаправлен в узел восходящего потока. Способ также

включает выбор свободной метки в узле нисходящего потока и информирование узла восходящего потока о том, что последующие пакеты, принадлежащие к определенному потоку, будут передаваться с выбранной присвоенной свободной меткой.

В другом варианте настоящее изобретение предусматривает способ для коммутации потока в первом узле, причем первый узел имеет линию связи нисходящего потока к второму узлу и линию связи восходящего потока к третьему узлу. Способ включает операции выполнения классификации потока в первом узле по первому пакету для определения того, принадлежит ли первый пакет к конкретному потоку, который должен быть перенаправлен в третий узел, выбора первой свободной метки в первом узле, информирования третьего узла, что последующие пакеты, принадлежащие к определенному потоку, будут передаваться с выбранной свободной меткой. Способ также включает выполнение классификации потока во втором узле по второму пакету для определения того, принадлежит ли второй пакет к определенному потоку, который должен быть перенаправлен к третьему узлу, выбора второй свободной метки во втором узле и информирования первого узла о том, что последующие пакеты, принадлежащие к определенному потоку, будут передаваться с выбранной второй присвоенной свободной меткой. Способ функционирует таким образом, что конкретный поток из восходящей линии связи может коммутироваться на уровне 2 первым узлом в нисходящую линию связи.

В соответствии с другим вариантом осуществления настоящее изобретение предусматривает базовый коммутационный блок в системе для передачи пакетов в сети. Базовый коммутационный блок включает аппаратные средства коммутации и контроллер, связанный с аппаратными средствами коммутации. Базовый коммутационный блок включает также программное обеспечение на материальном носителе, которое позволяет базовому коммутационному блоку осуществлять динамическое смещение между маршрутизацией пакетов посредством протокола IP уровня 3 и коммутацией на уровне 2 для оптимизации пропускной способности пакетного трафика.

В соответствии с еще одним вариантом осуществления настоящее изобретение обеспечивает коммутационный межсетевой блок (шлюзовой блок) в системе для передачи пакетов в сети. Система содержит базовый коммутационный блок, связанный с коммутационным шлюзовым блоком через канал связи. Коммутационный шлюзовой блок включает в себя контроллер шлюза и программное обеспечение. Контроллер шлюза содержит процессор, память и множество плат сетевых интерфейсов (NIC). Программное обеспечение на материальном носителе позволяет коммутационному шлюзовому блоку переадресовать поток пакетов к базовому коммутационному блоку, чтобы обеспечить динамическое смещение между маршрутизацией и коммутацией пакетов для оптимизации пропускной способности трафика.

В соответствии с еще одним вариантом осуществления настоящее изобретение обеспечивает средство коммутации в системе для передачи пакетов в сети. Система содержит базовый коммутационный блок, связанный со средством коммутации через канал связи, причем базовый коммутационный блок содержит контроллер и коммутатор. Средство коммутации включает в себя процессор, память и множество плат сетевых интерфейсов (NIC), причем конкретная одна из этих плат NIC обеспечивает канал связи и, по меньшей мере, одна из этих плат NIC имеет возможность соединения, по меньшей мере, с одним узлом сети. Коммутационное средство также содержит машиночитаемый программный код на материальном машиночитаемом носителе в памяти. Машиночитаемый программный код позволяет контроллеру базового коммутационного блока классифицировать поток и переадресовать этот поток пакетов из первого узла во второй узел в сети, а также позволяет контроллеру базового коммутационного блока выдавать команды средству коммутации осуществлять пересылку пакета данного потока из первого узла во второй узел через коммутатор. Соответственно функция пересылки пакетов не загружает контроллер базового коммутационного блока.

Указанные и другие варианты осуществления изобретения вместе с их признаками и преимуществами описаны детально в нижеприведенном тексте, иллюстрируемом чертежами.

Краткое описание чертежей

Фиг.1а- упрощенная диаграмма базового коммутационного блока системы, соответствующей возможному варианту осуществления изобретения; на фиг.1b показана упрощенная диаграмма коммутационного шлюзового блока системы, соответствующей другому варианту осуществления изобретения;

фиг. 1с - упрощенная диаграмма средства коммутации в системе, соответствующей еще одному варианту осуществления изобретения;

фиг. 2а-2с -упрощенные диаграммы примерных сетевых конфигураций в соответствии с вариантами осуществления изобретения;

фиг.3 - обобщенная блок-схема компьютерной системы, используемой в соответствии с вариантами осуществления изобретения;

фиг. 4 - обобщенная блок-схема АТМ коммутатора, используемого в соответствии с вариантами осуществления изобретения;

фиг. 5а - упрощенная диаграмма, иллюстрирующая в обобщенном виде процедуру инициализации в каждом системном узле, согласно возможному варианту осуществления изобретения;

фиг.5b - упрощенная диаграмма, иллюстрирующая работу системного узла;

фиг. 5с - упрощенная процедура, иллюстрирующая в обобщенном виде процедуру в средстве коммутации, когда пакет поступает в один из интерфейсов после инициализации;

фиг. 5d - упрощенная диаграмма, иллюстрирующая процедуру в контроллере

коммутации (с которым может быть связано посредством линии связи, по меньшей мере, одно средство коммутации, например, с использованием коммутатора контроллера коммутации), когда пакет поступает от средства коммутации на один из его интерфейсов по каналу, установленному по умолчанию, после инициализации;

фиг. 6а - диаграмма, иллюстрирующая в общем виде этапы, используемые для маркировки потока в системном узле;

фиг. 6b - диаграмма, иллюстрирующая в общем виде этапы, используемые для коммутации потока в базовом коммутационном узле;

фиг. 6с - диаграмма, иллюстрирующая в общем виде этапы, используемые для пересылки пакета в системном узле (или в узле коммутации);

фиг. 6d - диаграмма, иллюстрирующая в общем виде этапы, выполняемые в контроллере коммутации, при маркировке потока для пакетов, принимаемых от средства коммутации источника, в трех ситуациях;

фиг. 6е - диаграмма, иллюстрирующая в общем виде этапы, выполняемые в контроллере коммутации, при маркировке потока для пакетов, принимаемых от назначенного узла коммутации и предназначенных для интерфейса в назначенном средстве коммутации;

фиг.7а-7b - форматы идентификаторов потоков для потока типа 1 и для потока типа 2;

фиг. 8а - структура сообщения обобщенного протокола близости IFMP (протокол управления потоком Ipsilon), согласно одному из вариантов осуществления изобретения;

фиг.8b - обобщенный пакет протокола IP (в его современной версии IPv4) с полем данных переменной длины, в который может быть инкапсулировано сообщение IFMP;

фиг.8с - упрощенная диаграмма, иллюстрирующая работу системного узла после приема пакета с входящим сообщением протокола близости IFMP;

фиг.8d - упрощенная диаграмма, иллюстрирующая работу передаточного системного узла, если входящее сообщение протокола близости IFMP не является протоколом RSTACK;

фиг. 9а - структура обобщенного сообщения протокола перенаправления IFMP в соответствии с одним из вариантов осуществления изобретения;

фиг. 9b - обобщенная диаграмма, описывающая работу системного узла после приема сообщения протокола перенаправления IFMP;

фиг. 9с-9g - структуры элемента сообщения REDIRECT (перенаправление) элемента сообщения RECLAIM (восстановление), элемента сообщения RECLAIM ACK (подтверждение восстановления), элемента сообщения LABEL RANGE (интервал метки), элемента сообщения ERROR (ошибка) в теле сообщения 394 соответствующих сообщений протокола перенаправления IFMP;

фиг.10а - формат поля метки в линии передачи АТМ данных, согласно одному из вариантов осуществления изобретения;

фиг.10b -10е - иллюстрации соответственно пакетов, инкапсулированных

согласно протоколу IP, по умолчанию для потока типа 0, потока типа 1 и потока типа 2, согласно одному из вариантов осуществления настоящего изобретения;

фиг. 11a - формат инкапсулированного пакета протокола GSMP (обобщенный протокол управления коммутацией);

фиг. 11b - формат сообщения протокола близости GSMP;

фиг. 11c - упрощенная диаграмма, иллюстрирующая работу передающего узла после приема пакета с входящим сообщением протокола близости GSMP;

фиг. 11d - диаграмма состояний, иллюстрирующая работу передающего узла, если входящее сообщение протокола близости GSMP не является сообщением RSTACK;

фиг. 12 - формат обобщенного сообщения управления соединением протокола GSMP;

фиг. 13a - 13e - упрощенные диаграммы, иллюстрирующие работу приемного узла после приема сообщений "добавить переход", "исключить переход", "исключить дерево", "проверить дерево" и "удалить все" управления соединениями протокола GSMP соответственно;

фиг. 13f - формат сообщения "переместить корень" управления соединениями протокола GSMP;

фиг. 13g - упрощенная диаграмма, иллюстрирующая работу передающего узла после приема пакета с входящим сообщением "переместить корень" управления соединениями протокола GSMP;

фиг. 13h - формат сообщения "переместить переход" управления соединениями протокола GSMP;

фиг. 13i - упрощенная диаграмма, иллюстрирующая работу передающего узла после приема пакета с входящим сообщением "переместить переход" управления соединениями протокола GSMP;

фиг. 14 - формат сообщения управления портом протокола GSMP;

фиг. 15a - иллюстрация инкапсулированного пакета 1000 протокола IFMP-C;

фиг. 15b - общая структура типового сообщения IFMP-C 1002, которое может содержаться в поле 1006 сообщения IFMP-C инкапсулированного пакета 1000 протокола IFMP-C по фиг. 15a;

фиг. 16a - общая структура сообщения 1040 протокола близости IFMP-C, которое может содержаться в поле 1006 сообщения IFMP-C инкапсулированного пакета 1000 протокола IFMP-C по фиг. 15a;

фиг. 16b - диаграмма состояний, иллюстрирующая работу передающего узла (контроллера IFMP-C или исполнительного устройства (посредника) IFMP в трех возможных состояниях протокола близости IFMP-C;

фиг. 17a и 17b - структуры сообщений запроса и ответа перечня интерфейса IFMP-C соответственно;

фиг. 7c и 17d - структуры сообщений запроса и ответа очереди интерфейса IFMP-C соответственно;

фиг. 17e - структура сообщения 1170 запроса конфигурации интерфейса IFMP-C;

фиг. 18a - формат 1200 сообщений запроса "добавить переход" протокола IFMP-C и сообщений запроса "удалить

переход" IFMP-C;

фиг. 18b - поле 1240 преобразования данных для типа преобразования "усечение пакета" в сообщении запроса "добавить переход" протокола IFMP-C и сообщении запроса "удалить переход" IFMP-C по фиг. 18a;

фиг. 18c - формат 1250 сообщений ответа "добавить переход" протокола IFMP-C и сообщений ответа "удалить переход" IFMP-C;

фиг. 18d - структура сообщения 1260 запроса "удалить дерево" протокола IFMP-C;

фиг. 18e - структура сообщения 1300 запроса "переместить переход" протокола IFMP-C;

фиг. 19a - структура сообщения 1400 запроса "получить статистику дерева" протокола IFMP-C;

фиг. 19b - структура 1406 полей "данные дерева", иллюстрирующая используемые поля "данных дерева";

фиг. 20a и 20b - структура сообщения 1420 запроса "считать переход" протокола IFMP-C и сообщения 1430 ответа "считать переход" протокола IFMP-C соответственно;

фиг. 21a - структура сообщения 1440 запроса "информация узла" протокола IFMP-C;

фиг. 21b и 21c - структура сообщения 1460 запроса "статистика интерфейса" протокола IFMP-C и сообщения 1470 ответа "статистика интерфейса" протокола IFMP-C соответственно;

фиг. 21d - структура поля 1480 "статистика интерфейса" в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21c;

фиг. 21e - структура поля 1494 "общая статистика" в поле 1480 "статистика интерфейса" в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21c;

фиг. 21f - структура поля 1530 "специальная статистика" (для ATM интерфейса) в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21c;

фиг. 21g - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21c.

Детальное описание конкретных вариантов осуществления

Содержание

1. Общие сведения

2. Аппаратные средства системы

А. Аппаратные средства контроллера

В. Аппаратные средства коммутации

С. Пример осуществления аппаратных средств

3. Функциональные возможности программного обеспечения системы

А. Протокол IFMP и передача пакетов маркированного потока

В. Протокол GSMP

С. Протокол IFMP-C

4. Выводы

1. Общие сведения

Ниже раскрыты способ и устройство для передачи пакетов в сети. Эти способ и устройство могут быть, в частности, использованы при передаче с высокой пропускной способностью пакетов протокола

IP, которые обеспечивают передачу речевых сигналов, видеосигналов, сигналов данных в локальной сети (LAN), в сетях метрополий (MAN), в расширенных сетях (WAN), Интернет и т.п. Однако изобретение не ограничивается указанными типами сетей. Изобретение может использоваться в самых разнообразных системах, где требуется передача пакетов по сети.

Система, описываемая ниже, представляет собой систему динамической коммутации и маршрутизации. Эта система описывается в принципе как "система коммутации", однако следует иметь в виду, что данная система динамически обеспечивает как коммутацию на уровне 2 канала передачи данных, так и маршрутизацию и пересылку пакетов на сетевом уровне 3. Кроме того, "базовый коммутационный блок" системы также динамически обеспечивает как коммутацию на уровне 2, так и маршрутизацию и пересылку пакетов на уровне 3. "Межсетевой (шлюзовой) блок коммутации" системы служит в качестве устройства доступа, обеспечивая соединение существующей локальной сети и основной среды с сетью базовых коммутационных блоков. Аналогично шлюзовому блоку коммутации "исполнительное устройство (посредник) коммутации" также служит устройством доступа, предназначенным для обеспечения соединения существующей локальной сети и основной среды по меньшей мере с одним базовым коммутационным блоком. Как шлюзовой блок коммутации, так и базовый коммутационный блок имеют независимые средства управления перенаправлением потока, протоколами маршрутизации и принимают решения о маршрутизации независимо в отсутствие команд перенаправления потока, как описано ниже. Шлюзовой блок коммутации и базовый коммутационный блок являются поэтому равноправными. В противоположность этому исполнительное устройство коммутации, не имея независимых средств управления направлением потока, пересылает пакеты, основываясь на командах от базового коммутационного блока, действующего в качестве задающего блока по отношению к упомянутому посреднику. Работая в соответствии с командами от базового коммутационного блока, исполнительное устройство коммутации может пересылать пакеты, принятые от базового коммутационного блока, так что большая часть пакетов, направляемых базовым коммутационным блоком, может в данном случае направляться посредником в существующую локальную сеть или в основную операционную среду по интерфейсам посредника. Такие среды могут включать сети Ethernet, FastEthernet, FDDI (локальная сеть оптического стандарта), Gigabit Ethernet и другие типы локальных сетей. Поскольку такая пересылка пакетов выполняется посредником на базе команд пересылки пакетов, базовый коммутационный блок может иметь больше времени на выполнение других задач, связанных с выполнением протоколов маршрутизации, а также снизить время ожидания при пересылке пакетов. Выполнение задачи пересылки пакетов посредником сокращает нагрузку

контроллера коммутации базового коммутационного блока. Соответственно в некоторых ситуациях, когда средства независимого управления направлением потоков не требуются в определенном узле или если функциональные возможности базового коммутационного блока могут быть использованы более эффективно, указанный посредник оказывается пригодным для использования. Такой посредник может также использоваться в качестве экономичной замены шлюзового блока коммутации. Система совместима с протоколом Интернет (IP) в его текущей версии (IPv4), а также с перспективными версиями (например, IPv6). Система обеспечивает динамическое смещение между коммутацией и маршрутизацией пакетов в сети для обеспечения оптимального высокоскоростного распределения пакетов, позволяя избежать при этом "узких мест".

Как показано на фиг.1а, базовый коммутационный блок 1 системы коммутации, согласно возможному варианту осуществления настоящего изобретения, содержит машину 3 коммутации, контроллер 5 коммутации, программное обеспечение 7 системы, установленное в контроллере 5 коммутации. В частности, машина 3 коммутации использует обычные аппаратные средства коммутации асинхронного режима передачи (ATM). Разумеется, для реализации машины 3 коммутации в настоящем изобретении могут быть использованы и другие технологии, например быстродействующая пакетная коммутация, ретрансляция кадров, технология сети Gigabit Ethernet и другие, в зависимости от конкретного применения. В рассматриваемом варианте осуществления машина 3 коммутации представляет собой ATM коммутатор. Любое программное обеспечение, обычно связанное с ATM коммутатором, соответствующее уровню ATM адаптации типа 5 (AAL-5), полностью исключено. Таким образом, сигнализация, любой известный протокол маршрутизации, любой сервер эмуляции локальной сети или серверы определения адресов и т.п. исключены. Контроллер 5 коммутации представляет собой компьютер, имеющий адаптер ATM сети или плату сетевого интерфейса (NIC) 9, соединенные с машиной 3 коммутации посредством ATM канала связи 11. Программное обеспечение 7 системы установлено в базовом коммутационном блоке 1, более конкретно в компьютере, служащем в качестве контроллера 5 коммутации.

Машина 3 коммутации базового коммутационного блока 1 имеет множество физических портов 13, обеспечивающих соединение с различными устройствами, включая аппаратуру терминала данных (DTE), аппаратуру передачи данных (DCE), серверы, коммутаторы, шлюзы и т.п. Каждый из физических портов 13 может быть связан посредством ATM канала с устройством, оснащенным ATM адаптером или NIC, или с портом другого базового коммутационного блока, или с портом шлюзового блока коммутации, или с портом исполнительного механизма (посредника). Аппаратные средства ATM коммутации, предусматривающие машину 3 коммутации

базового коммутационного блока, действуют как уровень передачи данных (уровень 2 опорной модели OSI (протокол межсоединений открытых систем)).

Машина 3 коммутации служит для выполнения функции высокоскоростной коммутации, если это требуется базовым коммутационным блоком, как это определяется системным программным обеспечением 7. Функциональные возможности коммутации системы коммутации ограничены только аппаратными средствами, используемыми в составе машины 3 коммутации. Соответственно рассматриваемый вариант осуществления изобретения обеспечивает получение таких преимуществ ATM технологии, как высокое быстродействие, высокая пропускная способность и широкая полоса. Разумеется, другие технологии коммутации, такие как быстродействующая пакетная коммутация, ретрансляция кадров, технология сети Gigabit Ethernet и другие, также могут быть использованы в зависимости от конкретного применения.

В одном из вариантов осуществления настоящего изобретения контроллер 5 коммутации представляет собой компьютер, соединенный с аппаратными средствами 3 ATM коммутатора через канал ATM, а программное обеспечение системы установлено в компьютере. Кроме выполнения стандартных функций маршрутизации согласно протоколу IP, не требуя соединений, контроллер 5 коммутации также принимает решения о классификации пакетов на локальной основе.

Как показано на фиг.1b, шлюзовой блок коммутации 21 системы коммутации, соответствующей другому варианту осуществления настоящего изобретения, содержит контроллер 23 коммутации шлюза, системное программное обеспечение 25, установленное в контроллере 23 коммутации шлюза. Контроллер 23 коммутации шлюза содержит множество сетевых адаптеров или плат NIC 27 и ATM плату NIC 29. Аналогично контроллеру 5 коммутации базового коммутационного блока 1 контроллер 23 коммутации шлюза также представляет собой компьютер, оснащенный ATM платой NIC 29, имеющий системное программное обеспечение 25, установленное в компьютере. Как описано выше, шлюзовой блок коммутации 21 служит в качестве устройства доступа для обеспечения соединения существующей локальной сети и основной среды с сетью базовых коммутационных блоков. Соответственно платы NIC 27 могут быть различных типов, например 10BaseT Ethernet NICs, 100 BaseT Ethernet NICs, FDDI NICs и другие, а также комбинации перечисленных плат. Разумеется, использование конкретных плат NICs 27 зависит от типов существующих локальных сетей и основной операционной среды, в которой обеспечивается доступ посредством шлюзового блока коммутации 21. Ясно, что множество локальных сетей могут быть соединены со шлюзовым блоком коммутации 21. ATM плата NIC 29 позволяет шлюзовому блоку коммутации 21 соединяться через ATM канал с базовым коммутационным блоком 1. Разумеется, плата NIC 27 может представлять собой ATM плату NIC для

обеспечения соединения между шлюзовым блоком коммутации 21 и другим шлюзовым блоком коммутации.

Помимо базовых коммутационных блоков и шлюзовых блоков коммутации система, соответствующая настоящему изобретению, может содержать высокоэффективные главные компьютеры ("хосты"), рабочие станции или серверы, которые оснащены соответствующим образом. В частности, подсистема программного обеспечения может быть установлена на главном компьютере, рабочей станции или сервере, оснащенный соответствующей ATM платой NIC, чтобы обеспечить возможность такому главному компьютеру непосредственно соединяться с базовым коммутационным блоком.

Как показано на фиг.1c, исполнительное устройство (посредник) коммутации 901 в соответствии с еще одним вариантом осуществления изобретения представляет собой компьютер, оснащенный множеством сетевых адаптеров или плат NICs 903 для соединения с существующими локальными сетями и основной операционной средой, ATM платой NIC 905 для соединения с базовым коммутационным блоком 1 и соответствующим системным программным обеспечением 907, которое позволяет исполнительному устройству 901 коммутации пересылать пакеты по командам с базового коммутационного блока 1. Исполнительное устройство 901 служит в качестве устройства доступа для обеспечения соединения существующих локальных сетей и основной операционной среды, по меньшей мере, с одним базовым коммутационным блоком 1. Соответственно платы сетевых интерфейсов 903 могут быть одного и того же или различных типов, например платы 10BaseT Ethernet, 100 BaseT Ethernet, FDDI и другие, а также комбинации перечисленных плат. Разумеется, использование конкретных плат сетевых интерфейсов 903 зависит от типов существующих локальных сетей и основной операционной среды, в которой обеспечивается доступ посредством исполнительного устройства 901 коммутации. Ясно, что множество локальных сетей могут быть соединены с исполнительным устройством 901 коммутации. ATM плата 905 позволяет исполнительному устройству 901 коммутации соединяться через ATM канал с базовым коммутационным блоком 1. Разумеется, плата 905 выбирается надлежащим образом на базе конкретной технологии машины коммутации, в данном конкретном варианте - ATM, используемой в базовом коммутационном блоке 1.

Базовые коммутационные блоки, шлюзовые блоки коммутации, исполнительные устройства коммутации и системное программное обеспечение позволяют пользователям создать гибкие топологии сетей протокола IP, предназначенные для рабочих групп, организаций, операционных сред типа сетей WAN для достижения высокоэффективных масштабируемых решений существующих проблем перегрузок основных операционных сред рабочих групп. С использованием настоящего изобретения могут быть созданы разнообразные сетевые конфигурации, отличающиеся широкой полосой, высокой пропускной способностью, возможностью

взаимодействия компонентов систем. На фиг.2а-2с показаны конфигурации ряда сетей, которые могут быть реализованы в соответствии с настоящим изобретением. Разумеется фиг. 2а-2с иллюстрируют лишь приведенные для примера конфигурации, причем возможны многие другие примеры конфигураций.

На фиг.2а показана упрощенная диаграмма конфигурации локальной сети для рабочей (территориальной) группы, в которой базовый коммутационный блок 1 служит в качестве устройства централизованной пересылки пакетов по протоколу IP для всей локальной сети с несколькими шлюзовыми блоками 21 коммутации, обеспечивающими подключение к существующим локальным сетям. Базовый коммутационный блок 1 соединен с группой серверов, включающей три сервера 31_n (где $n=1...3$). Каждый сервер 31_n оснащен комплектом системного программного обеспечения и ATM платой NIC для обеспечения соединения с базовым коммутационным блоком 1 через соответствующие ATM каналы 33_n (где $n=1...3$), представляющие собой каналы OC-3 (155 Мб/с). С использованием серверов, непосредственно связанных с базовым коммутационным блоком 1 через высокоскоростные ATM каналы, обеспечивается ускорение пропускания пакетов для серверов с наиболее частым доступом. Базовый коммутационный блок 1 также соединен с тремя шлюзовыми блоками 21 коммутации через соответствующие ATM каналы 33_n (где $n=4...6$), также представляющие собой каналы OC-3. Первый шлюзовой блок 21 коммутации, соединенный с базовым коммутационным блоком 1 через канал 33_4 , также соединен с основной операционной средой 35_1 локальной сети, которая может представлять собой некоторый тип сети Ethernet или FDDI, через соответствующий канал 39_1 . Основная операционная среда 35_1 локальной сети связана с персональными компьютерами, терминалами, рабочими станциями 41 через соответствующие платы NICs 43. Аналогичным образом второй и третий шлюзовые блоки коммутации 21, соединенные с базовым коммутационным блоком 1 через каналы 33_5 , 33_6 , также соединены с основными операционными средами 35_2 , 35_3 локальных сетей через соответствующие каналы сети Ethernet или FDDI 39_2 , 39_3 . Конфигурация по фиг.2а таким образом позволяет пользователям непосредственно соединяться с различными локальными сетями для передачи без задержек потока трафика протокола IP, не создавая перегрузок, в соответствии с настоящим изобретением.

В качестве другого примера на фиг. 2b показана упрощенная диаграмма конфигурации рабочей группы. Фиг. 2b иллюстрирует высокоэффективную среду рабочей группы, в которой несколько главных компьютеров ("хостов") 45 соединены посредством ATM каналов 33_m с множеством базовых коммутационных блоков 1, которые соединены со шлюзовым блоком 21 коммутации, соединенным с локальной сетью 35, в которую включены пользовательские устройства 41. В этой конфигурации первый

базовый коммутационный блок 1 соединен с вторым базовым коммутационным блоком 1 посредством ATM канала 33_1 (155 Мб/с). Множество главных компьютеров 45 соединены с первым базовым коммутационным блоком 1 посредством соответствующего ATM канала 33_x со скоростью 155 Мб/с (где $x=2...5$) через соответствующую ATM плату NIC 47. Кроме того, множество главных компьютеров 45 соединены со вторым базовым коммутационным блоком 1 посредством соответствующего ATM канала 33_y со скоростью 25 Мб/с (где $y=8...10$) через соответствующую ATM плату NIC 49. Как описано выше, главные компьютеры 45 оснащены ATM платами NICs, с установленными в них комплектами системного программного обеспечения, что позволяет главным компьютерам, использующим протоколы TCP/IP, непосредственно соединяться с базовым коммутационным блоком. Первый и второй базовые коммутационные блоки 1 соединены с шлюзовым блоком 21 коммутации посредством ATM канала 33_6 (155 Мб/с) и 33_7 (25 Мб/с) соответственно.

Соединение первого и второго базовых коммутационных блоков 1 с шлюзовыми блоками 21 коммутации через канал 39 Ethernet или FDDI обеспечивает возможность пользователям главных компьютеров 45 осуществлять информационный обмен с пользовательскими устройствами 41, соединенными с локальной сетью 35. Пользовательские устройства 41 могут представлять собой персональные компьютеры, терминалы, рабочие станции, имеющие соответствующие платы NIC 43 для соединения к любой локальной сети 35 типа Ethernet или FDDI. Рабочая группа главных компьютеров тем самым оказывается интегрированной без каких-либо препятствий в остальную часть территориальной (например, университетской) сети.

На фиг.2с представлена конфигурация, использующая базовый коммутационный блок 1, несколько исполнительных устройств 911, 913, 915 коммутации и системный узел 916 (например, другой базовый коммутационный блок 1, шлюзовой блок коммутации или главный компьютер). Разумеется, в других конфигурациях могут использоваться дополнительные системные узлы и иные комбинации. На фиг.2с показано несколько исполнительных устройств 911, 913, 915 коммутации, каждое из которых имеет соответствующие интерфейсы с различными локальными сетями Ethernet 917_n (где $n=1...6$ в данном конкретном примере), каждое имеет подсоединенные пользовательские устройства (не показаны) и каждое из исполнительных устройств соединено посредством ATM каналов 919_m (где $m=1...3$ в данном конкретном примере) с базовым коммутационным блоком 1, который включает в себя контроллер 921 коммутации, соединенный ATM каналом 923 с машиной коммутации 925. Разумеется, локальные сети 917_n могут представлять собой 10BaseT или 100 BaseT Ethernet, FDDI, Gigabit Ethernet, другие типы сетей или комбинации этих типов сетей. В качестве пользовательских устройств, соединенных с локальной сетью

917_n, могут использоваться персональные компьютеры, терминалы, принтеры, серверы, рабочие станции и т.п., имеющие соответствующие платы NICs для соединения с локальной сетью 917_n. Системный узел 916 соединен с машиной коммутации 925 базового коммутационного блока 1 посредством ATM канала связи 919₄.

В общем случае контроллер коммутации 921 (фиг.2с) осуществляет управление исполнительными устройствами коммутации путем управления их интерфейсами (для передачи и приема пакетов) и путем управления ими в смысле обработки пакетов, принимаемых в конкретных потоках из специфических типов потоков. Конкретные типы потоков могут создаваться контроллером 921 коммутации согласно операциям по протоколу IFMP-C. Как упоминалось выше, контроллер 921 коммутации связан с коммутатором уровня канала (таким, как ATM коммутатор 925), который, в свою очередь, может быть связан с исполнительными устройствами коммутации (такими, как 911, 913, 915) и/или с другим системным узлом 916. В процессе инициализации контроллер 921 коммутации посылает пакеты протокола IFMP-C к исполнительным устройствам коммутации, что обеспечивает возможность контроллеру 921 коммутации определить конкретную конфигурацию (в смысле инсталлированных сетевых интерфейсов и т.п.) каждого исполнительного устройства коммутации. Контроллер 921 коммутации затем кондиционирует один или более сетевых интерфейсов 917_n, связанных с исполнительными устройствами коммутации, для начала приема пакетов. Контроллер 921 коммутации также устанавливает пакетную обработку в исполнительном устройстве коммутации для передачи определенных принимаемых пакетов к контроллеру 921 коммутации, в то время как другие принимаемые пакеты могут быть опущены (например, если они принимаются по протоколам, не обрабатываемым контроллером 921 коммутации). Если контроллер 921 коммутации обнаруживает, что поток может обрабатываться с помощью исполнительного устройства коммутации без вмешательства контроллера 921 коммутации, то контроллер 921 коммутации использует протокол IFMP-C для управления исполнительным устройством коммутации при обработке такого пакета (например, пропуске пакета, пересылке пакета с одного или более интерфейсов с использованием одного или более выходных форматов или с использованием других классов обслуживания для локальной пересылки пакета). С пересылкой пакета связано преобразование, применяемое для пакета (например, уменьшение "времени жизни" в пакете, обновление контрольной суммы заголовка IP, манипулирование заголовком для форматов других типов потока и т.п.). Ниже описаны дополнительные детали, относящиеся к взаимодействию исполнительных устройств коммутации, узла коммутации и контроллера коммутации (например, как показано в конфигурации по фиг.2с).

В соответствии с настоящим изобретением система добавляет полные

функциональные возможности маршрутизации по протоколу IP к аппаратным средствам ATM коммутации (либо альтернативной технологии в других вариантах) с использованием системного программного обеспечения, вместо любого имеющегося программного обеспечения управления ATM коммутатором, для управления ATM коммутатором. Таким образом, настоящая система способна осуществлять маршрутизацию по протоколу IP между сетевыми уровнями, если необходимо, и коммутацию уровней каналов передачи данных, если возможно, для создания высокоскоростной и высокопроизводительной пакетной передачи эффективным способом, избегая проблем перегрузок при маршрутизации.

С использованием протокола управления потоком Ipsilon (IFMP), который детально описан ниже, системное программное обеспечение дает возможность системному узлу (например, базовому коммутационному узлу, шлюзовому узлу коммутации или главному компьютеру/серверу/рабочей станции) классифицировать пакеты протокола IP как принадлежащие к потоку сходных пакетов, основываясь на определенных общих характеристиках. Поток представляет собой последовательность пакетов, передаваемых от конкретного источника к конкретному адресату (одиночному или групповому), которые связаны в смысле их маршрутизации или иной локальной обработки, которая может потребоваться. Настоящее изобретение обеспечивает возможность различной обработки различных типов пакетов в зависимости от типа пакета. Некоторые типы пакетов могут обрабатываться путем отображения их в индивидуальные ATM соединения с использованием ATM коммутатора для выполнения высокоскоростной коммутации пакетов. Потоки, например, такие, как несущие трафик реального времени, с требованиями качества обслуживания или имеющие длительное время выдержки, могут конфигурироваться для коммутации при появлении возможности. Другие типы потоков, например потоки короткой длительности или запросы в базы данных, обрабатываются путем IP маршрутизации без установления соединений. Конкретный поток пакетов может связываться с конкретной ATM меткой (т.е. ATM идентификатором виртуального маршрута (VPI) и идентификатором виртуального канала (VCI)). Предполагается, что виртуальные каналы являются однонаправленными, так что ATM метка входящего направления каждого канала принадлежит входному порту, с которым он соединен. Каждое направление передачи канала обрабатывается отдельно.

Разумеется, потоки, перемещающиеся в каждом направлении, обрабатываются системой отдельно аналогичным способом.

Классификация потока представляет собой решение, принимаемое в соответствии с локальной спецификой. Когда пакет протокола IP принимается системным узлом, системный узел передает пакет протокола IP по каналу, устанавливаемому по умолчанию. Узел также классифицирует пакет протокола IP, как принадлежащий к конкретному потоку, и соответственно принимает решение,

должны ли пакеты, принадлежащие к тому же потоку, предпочтительным образом коммутироваться непосредственно в ATM коммутаторе, или они должны продолжать пересылаться поэтапно с использованием программного обеспечения маршрутизации в узле. Если принято решение о коммутации потока пакетов, то поток должен сначала маркироваться. Для маркировки потока узел выбирает для этого потока доступную метку (VPI/VCI) входного порта, на который был принят пакет. Узел, который принял решение о маркировке потока, затем запоминает метку, идентификатор потока и время жизни и затем передает сообщение IFMP REDIRECT (переадресация) по восходящей линии к предыдущему узлу, от которого поступил пакет. Идентификатор потока содержит набор полей заголовка, которые характеризуют поток. Время жизни определяет длительность времени, в течение которого действует переадресация. Если только состояние потока не было обновлено, связь между потоком и меткой удаляется по истечении указанного времени жизни. Истечение времени жизни до обновления состояния потока приводит к тому, что последующие пакеты, принадлежащие потоку, будут передаваться по каналу передачи, устанавливаемому по умолчанию, между соседними узлами. Состояние потока обновляется передачей по восходящей линии сообщения "переадресация", имеющего те же самые метку и идентификатор потока, что и первоначальные, но имеющего другое время жизни. Сообщение "переадресация" запрашивает узел восходящей линии передавать все последующие пакеты, которые имеют характеристики, совпадающие с характеристиками, указанными в идентификаторе потока, по виртуальному каналу, указанному меткой. Решение о переадресации представляет собой локальное решение, обрабатываемое узлом нисходящей линии. Соответственно даже если узел нисходящей линии запрашивает переадресацию конкретного потока пакетов, узел восходящей линии может принять решение о принятии или игнорировании запроса на переадресацию. Кроме того, для сообщения "переадресация" не передается подтверждение приема. Напротив, первый пакет, приходящий по новому виртуальному каналу, служит для индикации того, что запрос переадресации принят.

Системное программное обеспечение также использует различные процедуры формирования пакетов протокола IP, которые принадлежат к маркированным потокам в линии передачи ATM данных, в зависимости от различных типов потоков. В рассматриваемом варианте осуществления используются четыре типа формирования пакетов.

Помимо протокола IFMP, системное программное обеспечение использует другой протокол - общий протокол управления коммутацией (GSMP) - для установления связи по линии передачи ATM данных между контроллером коммутации и ATM коммутатором базового коммутационного блока системы и при этом обеспечивая возможность коммутации на уровне 2, если возможно, и маршрутизации по протоколу IP на уровне 3 и пересылки пакетов, если

необходимо. В частности, протокол GSMP представляет собой универсальный асимметричный протокол управления ATM коммутатором. Это означает, что контроллер коммутации действует в качестве ведущего, а ATM коммутатор - в качестве подчиненного. Протокол GSMP реализуется в виртуальном канале, установленном при инициализации в линии передачи ATM данных между контроллером коммутации и ATM коммутатором. Один контроллер коммутации может использовать множество конкретизации понятий GSMP по различным виртуальным каналам для управления множеством ATM коммутаторов. В протокол GSMP также включен протокол близости GSMP, который используется для синхронизации состояния в линии передачи ATM данных между контроллером коммутации и ATM коммутатором, для определения данных идентификации узла на другом конце линии связи и для обнаружения изменений в идентификации этого узла.

Протокол GSMP позволяет контроллеру коммутации устанавливать и освобождать соединения через ATM коммутатор, добавлять и исключать "листья" (элементы нижнего уровня иерархии в древовидной структуре иерархии) в соединении типа "от точки к множеству точек", управлять портами коммутации, запрашивать информацию конфигурации и запрашивать статистику. Протокол GSMP также позволяет ATM коммутатору информировать контроллер коммутации о событиях, например отказе линии связи.

Предполагается, что коммутатор имеет множество портов, причем каждый порт представляет собой комбинацию входного и выходного портов. Элементы ATM данных (cell) поступают от внешней линии связи по входящим виртуальным каналам на входной порт и передаются от ATM коммутатора во внешнюю линию связи по исходящим виртуальным каналам с выходного порта. Как упоминалось выше, виртуальные каналы в порте или линии связи идентифицируются по их меткам виртуальных маршрутов и виртуальных каналов (VPI/VCI). Соединение с использованием виртуального канала через ATM коммутатор формируется путем соединения входящего виртуального канала (или корня) с одним или более исходящими виртуальными каналами (или ветвями). Соединения с использованием виртуальных каналов идентифицируются по входному порту, на который они приходят, и по метке VPI/VCI их входящего виртуального канала. В коммутаторе каждый порт имеет таблицу перекодировки аппаратных средств, индексированную метками VPI/VCI входящих ATM элементов данных, причем записи в таблице контролируются процессором локального управления в коммутаторе.

В случае протокола GSMP каждое соединение с использованием виртуального канала может быть установлено с определенным качеством обслуживания, путем присвоения приоритета при его установлении. Для соединений с использованием виртуальных каналов, совместно использующих один и тот же выходной порт, ATM элемент данных в соединении с более высоким приоритетом будет передан коммутатором с более высокой

вероятностью, чем ATM элемент данных в соединении с более низким приоритетом, если они оба присутствуют в коммутаторе в одно и то же время. Число приоритетов, поддерживаемых каждым портом коммутатора, может быть определено из сообщения конфигурирования порта. Ясно, что различные коммутаторы могут поддерживать групповую передачу различными путями. Например, коммутатор может иметь пределы по числу ветвей для реализации группового соединения, пределы по количеству поддерживаемых групповых соединений, пределы по количеству различных значений VPI/VCI, которые могут быть присвоены выходным ветвям группового соединения, и/или поддерживать только одну ветвь конкретного группового соединения на одном и том же выходном порте. Если необходимо, могут быть определены коды отказов для соответствующих ситуаций.

Коммутатор присваивает 32-битовые номера портов для описания портов коммутатора. Номер порта может быть структурирован для включения субполей, относящихся к физической структуре коммутатора (например, стойка, слот, порт). Каждый порт коммутатора также поддерживает номер сеанса связи порта, присвоенный коммутатором. Номер сеанса связи порта остается тем же самым при постоянном рабочем состоянии порта. Однако если порт возвращается в рабочее состояние после его отказа или недоступности или цикла подачи питания, номер сеанса связи порта будет изменен. Номера сеансов связи портов присваиваются с использованием некоторой формы случайного числа и позволяют контроллеру коммутации обнаруживать отказы линий связи и поддерживать состояние синхронизации.

Помимо протоколов IFMP и GSMP системное программное обеспечение в некоторых вариантах осуществления изобретения также использует еще один протокол - протокол управления потоком Ipsilon для клиентов (IFMP-C), который описан детально ниже, для установления информационного обмена по линии связи между контроллером коммутации базового коммутационного блока и исполнительным устройством коммутации, чтобы распределять пересылку пакетов уровня 3 к исполнительным устройствам коммутации, если это желательно. В частности, протокол GFMP-C представляет собой универсальный асимметричный протокол для управления исполнительным устройством коммутации. Таким образом, контроллер коммутации действует в качестве ведущего узла, а исполнительное устройство коммутации - в качестве подчиненного. При использовании протокола IFMP-C интерфейсы на исполнительном устройстве коммутации сходны с интерфейсами, локально присвоенными контроллеру коммутации, так что контроллер коммутации/исполнительное устройство коммутации с внешней стороны представляется подобным системному узлу. В принципе протокол IFMP-C используется в виртуальном канале, установленном при инициализации в линии связи между контроллером коммутации и исполнительным устройством коммутации. Один контроллер коммутации может использовать множество

конкретизаций понятий протокола IFMP-C для отдельных виртуальных каналов для управления множеством исполнительных устройств коммутации. При запуске системы контроллер коммутации запускает приемник протокола IFMP-C в каждом ATM интерфейсе (приемнику поставлен в соответствие устанавливаемый по умолчанию виртуальный канал ATM интерфейса), связанном с контроллером коммутации, и исполнительное устройство коммутации начинает передачу периодических сообщений синхронизации (SYN) по установленному по умолчанию виртуальному каналу. Когда контроллер коммутации принимает сообщение SYN от исполнительного устройства коммутации, контроллер коммутации запускает протокол близости IFMP-C, который включен в протокол IFMP-C. Предназначенный для использования на каждом конце линии связи протокол близости IFMP-C используется для синхронизации состояния в линии связи между контроллером коммутации и исполнительным устройством коммутации, чтобы обнаружить идентификационные данные узла на другом конце линии связи и обнаружить изменения в идентификации этого узла. После того как протокол близости IFMP-C установит, что каждая сторона линии связи синхронизирована с другой стороной, каждая сторона линии связи будет иметь номер экземпляра записи, идентифицирующий другую сторону линии связи.

После завершения синхронизации протокол IFMP-C позволяет контроллеру коммутации определить, какие порты или интерфейсы (и их атрибуты) доступны в исполнительном устройстве коммутации, и конфигурировать каждый интерфейс таким образом, что он может быть использован для пересылки пакетов. Исполнительное устройство коммутации содержит множество портов или интерфейсов, причем каждый интерфейс или порт является комбинацией входного порта и выходного порта. После того как интерфейсы определены и конфигурированы, протокол IFMP-C используется для создания, модифицирования и исключения ветвей передачи. Каждая ветвь передачи состоит из входных данных и выходных данных. В каждом исполнительном устройстве коммутации каждый интерфейс имеет таблицу перекодировки аппаратных средств, индексированную входными данными/выходными данными входящего пакета, а записи в таблице контролируются локальным процессором управления в исполнительном устройстве коммутации. Входные данные включают в себя различные элементы и компоненты информации (например, входной интерфейс, предшествование, входные флаги, данные ключа, маску ключа в соответствии с конкретным вариантом осуществления), причем каждый элемент информации вносит вклад во входную информацию. Если какие-либо компоненты входных данных меняются, то пакет рассматривается как имеющий другую входную запись пересылки. Выходные данные включают в себя различные элементы и компоненты информации (например, выходной интерфейс, длину удаления, преобразование,

данные преобразования, данные заголовка, качество типа обслуживания, данные качества обслуживания в соответствии с конкретным вариантом осуществления), которые описывают то, каким образом должны пересылаться пакеты, имеющие согласованные входные данные. Возможно, что одна входная запись будет иметь более одной выходной записи. Когда пакет приходит на интерфейс исполнительного устройства коммутации, исполнительное устройство коммутации осуществляет поиск во входных записях, связанных с входным интерфейсом.

Поиск записей осуществляется от низших к высшим по старшинству. Если обнаружено совпадение входных записей, то информация о выходных ветвях используется для пересылки пакета.

В случае протокола IFMP-C управление аппаратными средствами на уровне линии связи (например, открытие виртуальных каналов и добавление фильтров адресов аппаратных средств для сети Ethernet) сохраняется за исполнительным устройством коммутации. Если входная маска ключа включает биты адреса на уровне линии связи, то исполнительное устройство коммутации должно гарантировать прием этих адресов. Если маска не включает информации адресации на уровне линии связи, то исполнительное устройство коммутации не должно настраивать фильтр. Исполнительное устройство коммутации должно таким образом управлять фильтрацией на уровне линии связи наиболее эффективным образом для своих аппаратных средств, а контроллер коммутации должен включать достаточное количество информации уровня линии связи в ключ, чтобы надлежащим образом осуществлять фильтрацию пакетов. Контроллер коммутации управляет состоянием исполнительного устройства коммутации для разнородного и группового разнородного режимов передачи так, чтобы исполнительное устройство коммутации не пыталось ненадлежащим образом оптимизировать кодовый маршрут вне пределов желательного режима.

Протоколы IFMP, GSMP и IFMP-C детально описаны ниже в соответствии с конкретным вариантом осуществления настоящего изобретения.

2. Аппаратные средства системы

A. Аппаратные средства контроллера

На фиг.3 представлена блок-схема типовой компьютерной системы 51, которая может быть использована в качестве контроллера 5 коммутации в базовом коммутационном блоке 1 (как показано на фиг.1a) для реализации системного программного обеспечения согласно настоящему изобретению. На фиг.3 также показан пример компьютерной системы, которая может быть использована в качестве контроллера 23 шлюзового блока коммутации в шлюзовом блоке 21 коммутации (как показано на фиг.1b) для реализации системного программного обеспечения согласно настоящему изобретению, а также служащего примером типового компьютера, который может быть использован в качестве главного компьютера/сервера/рабочей станции с комплектом системного программного обеспечения. Разумеется, другие элементы, такие как монитор, экран,

клавиатура, добавляются для главного компьютера. Как показано на фиг.3, компьютерная система 51 содержит такие подсистемы как центральный процессор 69, системная память 71, контроллер ввода/вывода 73, стационарный диск 79, сетевой интерфейс 81, ПЗУ 83. Компьютерная система 51 в случае, если она является главным компьютером ("хостом"), может дополнительно содержать монитор 53, клавиатуру 59, адаптер дисплея 75, съемный диск 77. Стрелки 85 представляют архитектуру системной шины компьютерной системы 51. Однако эти стрелки лишь иллюстрируют схему взаимных соединений, обеспечивающих связь между подсистемами. Например, локальная шина может использоваться для соединения центрального процессора 69 с системной памятью 71 и ПЗУ 83. Другие компьютерные системы, пригодные для использования в настоящем изобретении, могут включать дополнительные подсистемы или меньшее количество подсистем. Например, другая компьютерная система может включать более одного процессора 69 (мультипроцессорная система) или кэш-память (сверхоперативную память).

В возможном варианте осуществления изобретения компьютер, используемый в качестве контроллера коммутации, представляет собой стандартную машину на базе центрального процессорного блока типа Intel, оснащенного стандартной шиной PCI (соединения периферийного оборудования), а также адаптером ATM сети или платой сетевого интерфейса (NIC). Компьютер соединен с ATM коммутатором посредством ATM линии связи со скоростью передачи 155 Мбит/с, использующей плату интерфейса ATM сети. В данном варианте осуществления системное программное обеспечение установлено на стационарном диске 79, который представляет собой накопитель на жестких дисках компьютера. Специалистам в данной области техники ясно, что системное программное обеспечение может храниться на ПЗУ на компакт-диске (CD-ROM), на гибком диске, на магнитной ленте или иных материальных носителях, которые могут хранить машиночитаемые коды.

Компьютерная система 51, показанная на фиг.3, представляет собой лишь возможный пример компьютерной системы, пригодной для использования (в качестве контроллера коммутации базового коммутационного блока, контроллера шлюзового блока коммутации или главного компьютера/сервера/рабочей станции) в соответствии с настоящим изобретением. Кроме того, фиг.3 иллюстрирует пример компьютерной системы, установленной с использованием по меньшей мере части системного программного обеспечения (для обеспечения режима работы в соответствии с протоколом IFMP-C), которая может быть использована в качестве исполнительного устройства коммутации 901 (как показано на фиг.1c). Следует иметь в виду, что системное программное обеспечение для протоколов маршрутизации не обязательно устанавливается в компьютерной системе, служащей в качестве исполнительного устройства 901 коммутации, и поэтому данная часть системного программного обеспечения может реализовываться на встроеном

устройстве. Соответственно стационарный диск 79 может быть опущен в компьютерной системе, используемой в качестве исполнительного устройства 901 коммутации, что приводит к меньшим затратам на аппаратуру для сетей, которые могут использовать исполнительные устройства 901 коммутации вместо шлюзовых блоков коммутации. Для специалистов в данной области техники очевидны и другие конфигурации подсистем, пригодных для использования в соответствии с настоящим изобретением. Кроме того, шлюзовой блок коммутации может быть оснащен множеством других плат сетевых интерфейсов (NIC) для обеспечения соединения с другими типами локальных сетей. Другие платы NIC или альтернативные адаптеры для других типов основных операционных сред могут быть использованы в шлюзовом блоке коммутации. Например, могут использоваться платы SMC 10M/100M Ethernet или FDDI. Без каких-либо ограничений объема изобретения. Табл. 1 представляет перечень коммерчески доступных компонентов, которые используются при эксплуатации контроллера в соответствии с вышеописанными вариантами осуществления. Для специалистов в данной области техники должно быть очевидно, что компоненты, приведенные в табл.1, являются лишь характерными компонентами для реализации изобретения и приведены для пояснения выполнения устройства в соответствии с конкретным вариантом осуществления изобретения. Различные другие компоненты, известные специалистам в данной области техники, могут быть использованы взамен, либо в комбинации или отдельно с другими компонентами.

В. Аппаратные средства коммутации

Как описано выше, аппаратные средства коммутации ATM предусматривают исполнительное устройство коммутации или базовый коммутационный блок. ATM коммутатор использует независимые от поставщика аппаратные средства ATM коммутации. Однако машина ATM коммутации, соответствующая настоящему изобретению, не основывается на обычном, ориентированном на соединения программном обеспечении ATM маршрутизации и сигнализации (SSCOP, Q.2931, UNI 3.0/3.1, P-NNI). Напротив, протоколы и программное обеспечение режима ATM полностью не используются, а базовый коммутационный блок основывается на системном программном обеспечении для управления машиной ATM коммутации. Системное программное обеспечение более детально описано ниже.

Отдельно доступные ATM компоненты могут быть собраны в типовую архитектуру ATM коммутатора. Например, на фиг.4 представлена обобщенная блок-схема архитектуры ATM коммутатора 3 (пример иллюстрирует 16-портовый коммутатор), который может быть использован в качестве аппаратного средства коммутации базового коммутационного узла в соответствии с возможным вариантом осуществления изобретения. Однако коммерчески доступные ATM коммутаторы также могут работать в качестве машины коммутации базового коммутационного блока в соответствии с

другими вариантами осуществления настоящего изобретения. Основные функциональные компоненты аппаратных средств коммутации 3 включают переключающий сердечник, микроконтроллерный комплекс, приемопередающий подузел. В принципе, переключающий сердечник выполняет коммутацию на уровне 2, микроконтроллерный комплекс обеспечивает системное управление ATM коммутатором, и приемопередающий подузел обеспечивает интерфейс и базовую передачу и прием сигналов от физического уровня. В данном примере переключающий сердечник основан на комплекте микросхем ATM коммутатора MMC Networks ATMS 2000, который включает в себя "белую микросхему" 100, "серую микросхему" 102, микросхемы MBUF 104, микросхемы интерфейса порта (PIF) 106 и общую память данных 108. Переключающий сердечник может дополнительно включать детектор активности виртуального канала 110, функциональный блок 112 отбрасывания раннего пакета. Также включены счетчики пакетов, не показанные на чертеже. Белая микросхема 100 обеспечивает управление конфигурацией и статус. Помимо обмена данными с белой микросхемой 100, относящимися к статусу и управлению, серая микросхема 102 обеспечивает прямую адресацию и перенос данных с использованием таблиц коммутации. Микросхемы MBUF 104 обеспечивают перемещение трафика элементов данных между микросхемами интерфейса порта 106 и памятью общих данных 108. Память общих данных 108 используется для буферизации элементов данных в коммутаторе. Микросхемы интерфейса порта 106 управляют переносом данных между микросхемами MBUF и аппаратными средствами портов коммутации. Детектор активности 110 виртуального канала, включающий в себя элемент памяти, обеспечивает информацию о каждом действующем виртуальном канале. Функциональный блок 112 отбрасывания ранних пакетов обеспечивает возможность отбрасывать некоторые ATM элементы данных по мере необходимости. Счетчики пакетов обеспечивают в коммутаторе средство подсчета всех пакетов, проходящих через все входные и выходные порты. Шины 114, 115, 117 и 118 обеспечивают интерфейс между различными компонентами коммутатора. Микроконтроллерный комплекс включает в себя центральный процессорный блок 130, динамическую память со случайным доступом 132, ПЗУ 134, флэш-память 136, контроллер 138 динамической памяти со случайным доступом, сдвоенные универсальные асинхронные приемопередающие порты 140 и 142 и внешний таймер 144. Центральный процессорный блок 130 действует в качестве микроконтроллера. ПЗУ 134 действует в качестве ПЗУ локальной загрузке и включает в себя полностью отображение кодов переключения, функциональные средства операционной системы базового нижнего уровня и диагностики. Динамическая память со случайным доступом 132 обеспечивает обычные функции ОЗУ, а контроллер 138 (который может быть

реализован как перепрограммируемое устройство на вентильной матрице и т.п.) обеспечивает управление динамической памятью со случайным доступом 132. Флэш-память 136 используется микроконтроллером для управления пересмотром аппаратных средств, идентификации серийного номера и различных контрольных кодов для целей производства и отслеживания. Сдвоенные универсальные асинхронные приемопередающие порты 140 и 142 обеспечивают интерфейсы к коммуникационным ресурсам для диагностики, контроля и иных целей. Внешний таймер 144 осуществляет прерывания для центрального процессорного блока, как это требуется. Приемопередающий блок включает в себя физические интерфейсы 146, размещенные между микросхемой 106 и физическими приемопередатчиками (не показаны). Интерфейсы 146 выполняют обработку потока данных и воплощают физический уровень ATM. Разумеется, компоненты коммутатора могут быть реализованы на печатной плате, которая может быть размещена в стойке для ее монтажа или установлена в настольном варианте, в зависимости от используемого типа шасси.

Не ограничивая объем изобретения, табл.2 представляет перечень коммерчески доступных компонентов, которые могут использоваться для эксплуатации машины коммутации в соответствии с вышеописанными вариантами осуществления. Для специалистов в данной области техники очевидно, что компоненты, приведенные в табл.2, даны лишь в качестве характерных компонентов, которые могут быть использованы в связи с изобретением, и приведены лишь для примера, иллюстрирующего сборку устройства, соответствующего конкретному варианту осуществления изобретения. Различные другие компоненты, известные специалистам в данной области техники, могут быть использованы взамен, либо в комбинации или отдельно с другими компонентами. Разумеется, как упоминалось выше, машины коммутации, использующие технологии, отличные от ATM (например, ретрансляция кадров, высокоскоростная пакетная коммутация, Gigabit Ethernet), могут использовать соответствующие компоненты.

3. Функциональные средства системного программного обеспечения

Как описано в общем виде выше, протокол IFMP представляет собой протокол для передачи команд соседнему узлу присвоить метку уровня 2 конкретному потоку пакетов. Поток представляет собой последовательность пакетов, передаваемых от конкретного источника к конкретному адресату (адресатам), которые связаны их маршрутизацией и требуемой логической обработкой. Метка определяет виртуальный канал и обеспечивает доступ к кашируемой информации маршрутизации для данного потока. Метка также обеспечивает то, что пакеты, принадлежащие определенному потоку, коммутируются на уровне 2, а не маршрутизируются на уровне 3. Это означает, что если восходящая и нисходящая линии связи переадресуют поток в конкретный узел

сети, то данный конкретный узел может коммутировать этот поток на уровне линии передачи данных, а не маршрутизирует и пересылает поток на сетевом уровне.

На фиг.5a представлена упрощенная диаграмма, иллюстрирующая в общем виде процедуру инициализации в каждом системном узле в соответствии с одним из вариантов осуществления изобретения. После запуска системы на этапе 160 каждый системный узел на этапе 162 устанавливает по умолчанию виртуальные каналы на всех портах. Затем на этапе 164 каждый системный узел ожидает поступления пакетов на любой из портов.

На фиг. 5b представлена упрощенная диаграмма, иллюстрирующая работу системного узла, осуществляющего динамические смещения между маршрутизацией на уровне 3 и коммутацией на уровне 2 в соответствии с настоящим изобретением. После инициализации на этапе 166 пакет поступает на порт системного узла. Если пакет принят по виртуальному каналу, установленному по умолчанию (этап 168), то системный узел выполняет классификацию потока для пакета на этапе 170. Классификация потока связана с определением того, принадлежит ли пакет к конкретному типу потока. На этапе 172 системный узел определяет, должен ли поток, к которому принадлежит пакет, предпочтительно коммутироваться. Если системный узел определил, что поток должен коммутироваться, то системный узел маркирует поток на этапе 174 и затем на этапе 176 пересылает пакет. После пересылки пакета системный узел ожидает прибытия пакета на этапе 182. Как только пакет поступает, системный узел возвращается к этапу 176. Если на этапе 168 системный узел определит, что пакет не поступил по виртуальному каналу, установленному по умолчанию, то системный узел не выполняет классификацию потока на этапе 170. Если пакет поступает по другому виртуальному каналу, то поступивший пакет принадлежит к потоку, который уже был маркирован раньше. Соответственно, если поток уже маркирован в нисходящей линии (этап 178), то системный узел коммутирует поток на этапе 180. Коммутация потока предусматривает установление соединения в коммутаторе между меткой восходящей линии связи и меткой нисходящей линии связи. После коммутации потока на этапе 180 системный узел на этапе 176 передает пакет в нисходящую линию. Если поток не маркирован в нисходящей линии (этап 178), то системный узел не коммутирует поток, а передает пакет в нисходящую линию на этапе 176. Следует иметь в виду, что только системный узел, являющийся базовым коммутационным блоком, выполняет этап 180. Другие системные узлы (например, шлюзовой блок коммутации или главный компьютер) работают так, как показано на фиг.5b, но не выполняют этапа 180, поскольку результат этапа 178 не предназначается ни для шлюзового блока коммутации, ни для главного компьютера (так как эти типы системных узлов не имеют нисходящей линии связи). На фиг.5c и 5d показаны упрощенные диаграммы, которые иллюстрируют работу контроллера коммутации и исполнительного

устройства коммутации, связанного с контроллером коммутации посредством линии связи соответственно в соответствии с настоящим изобретением. Отметим, что исполнительное устройство коммутации в общем случае следует процедуре инициализации, представленной на фиг.5a. Фиг. 5c в общем виде иллюстрирует процедуру в исполнительном устройстве коммутации, когда пакет поступает (этап 1600) на один из его интерфейсов после того, как инициализация завершена. Если пакет не принят по виртуальному каналу, установленному по умолчанию (определяется на этапе 1602), то исполнительное устройство коммутации получает доступ к дереву, связанному с конкретным каналом на этапе 1604. Если пакет не поступает по каналу, установленному по умолчанию, то это означает, что пакет принадлежит потоку, который уже был маркирован и поток уже коммутировался. Исполнительное устройство коммутации продолжает пересылать пакет (этап 1606) соответствующим образом и затем ожидает прихода другого пакета (этап 1608). Однако если пакет принят по виртуальному каналу, установленному по умолчанию (определяется на этапе 1602), то исполнительное устройство коммутации осуществляет поиск в его таблице переходов для отыскания на этапе 1610 согласованной входной ветви. Если согласованная входная ветвь не найдена (на этапе 1612), то исполнительное устройство коммутации передает пакет к контроллеру коммутации на этапе 1614 и ожидает поступления другого пакета (этап 1616). Если согласованная входная ветвь найдена (на этапе 1612), то исполнительное устройство коммутации передает пакет, как определено на этапе 1618. Затем исполнительное устройство коммутации проверяет, не определен ли режим "передачи управления вниз" ("провалиться") для данного пакета (этап 1620). Как пояснено ниже, режим "передачи управления вниз" означает, что исполнительное устройство коммутации должно продолжать поиск в таблице переходов для отыскания согласованной входной ветви на следующем по старшинству уровне, который согласован с этой входной ветвью после того, как пакет передан. Если режим "передачи управления вниз" не определен (этап 1620), то исполнительное устройство коммутации просто ожидает прихода следующего пакета (этап 1622). Однако если режим "передачи управления вниз" определен (этап 1620), то исполнительное устройство коммутации продолжает поиск в таблице переходов для отыскания согласованной входной ветви на следующем по старшинству уровне (этап 1624). После этапа 1624 исполнительное устройство коммутации определяет, найдена ли согласованная входная ветвь на следующем по старшинству уровне (этап 1626). Если не найдена, то исполнительное устройство коммутации просто ожидает прихода следующего пакета (этап 1622). Однако если она найдена, то исполнительное устройство коммутации продолжает после этапа 1626 направлять пакет в соответствии с тем, как это определено (этап 1618), причем процедура продолжается от этапа 1620.

На фиг. 5d представлена процедура в

контроллере коммутации (с которым может быть связано по меньшей мере одно исполнительное устройство коммутации с помощью канала связи), когда пакет поступает (этап 1650) от исполнительного устройства коммутации на один из его интерфейсов по каналу, установленному по умолчанию, после того, как завершена инициализация. Это означает, что на фиг. 5d показана процедура в контроллере коммутации после того, как пакет передан в контроллер коммутации (этап 1614 на фиг.5c). После того как пакет поступает (этап 1650) от исполнительного устройства коммутации, контроллер коммутации выполняет классификацию потока по пакету на этапе 1652. Как отмечено выше, классификация потока связана с определением того, принадлежит ли пакет к конкретному типу потока. Исходя из этапа 1652, контроллер коммутации определяет на этапе 1654, следует ли коммутировать поток, к которому принадлежит данный пакет. Если контроллер коммутации определяет на этапе 1654, что поток не следует коммутировать, то контроллер коммутации не коммутирует поток, а просто пересылает пакет (этап 1656) и затем ожидает следующего пакета (этап 1658). Если контроллер коммутации определяет на этапе 1654, что поток должен быть коммутирован, то контроллер коммутации маркирует поток на этапе 1660 и приступает к пересылке пакета (этап 1656) и затем ожидает следующего пакета (этап 1658).

На фиг. 6a представлена диаграмма, иллюстрирующая этапы, связанные с маркировкой потока в восходящей линии связи системного узла (или узла коммутации), как показано этапом 174 маркировки потока на фиг.5b. Для системного узла, который представляет собой шлюзовой блок коммутации или главный компьютер, системный узел маркирует поток, как показано для этапов 190, 192, 200 и 202 на фиг.6a. Когда начинается этап маркировки потока (этап 190), системный узел выделяет свободную метку x в восходящей линии на этапе 192. Системный узел затем пересылает сообщение "переадресация" протокола IFMP в восходящую линию на этапе 200 (как показано пунктирной линией 193). Системный узел затем направляет пакет на этапе 202. Для системного узла, который представляет собой базовый коммутационный блок, маркировка потока также иллюстрируется этапами 194, 196 и 198. Когда начинается этап маркировки потока (этап 190), базовый коммутационный блок выбирает свободную метку x в восходящей линии на этапе 192. Контроллер коммутации базового коммутационного блока затем выбирает временную метку x' в порте управления контроллера коммутации на этапе 194. На этапе 196 контроллер коммутации пересылает аппаратному средству коммутации сообщение протокола GSMP для отображения метки x в восходящей линии связи в метку x' в порте управления. Контроллер коммутации затем ожидает на этапе 198, пока не будет принято сообщение подтверждения приема протокола GSMP от аппаратного средства коммутации, которое указывает, что отображение было успешным. После приема подтверждения базовый

коммутационный блок посылает сообщение "переадресация" протокола IFMP по восходящей линии на этапе 200. После этапа 200 системный узел возвращается к этапу 176, как показано на фиг.5b.

Фиг. 6b представляет диаграмму, иллюстрирующую этапы, осуществляемые при коммутации потока в базовом коммутационном блоке, как показано этапом 180 коммутации потока на фиг.5b. Как упомянуто выше, только системные узлы, представляющие собой базовые коммутационные блоки, могут выполнять этап коммутации потока. Когда на этапе 210 начинается процедура коммутации потока, контроллер коммутации в базовом коммутационном блоке передает на этапе 212 сообщение протокола GSMP для отображения метки x в восходящей линии связи в метку y в нисходящей линии связи. Метка y представляет собой метку, которую узел, размещенный в нисходящей линии относительно базового коммутационного блока, присвоил потоку. Разумеется, этот узел нисходящей линии маркировал поток таким образом, как рассмотрено со ссылками на фиг.5b и 6a, причем свободная метка y выбирается на этапе 192. После этапа 212 контроллер коммутации в базовом коммутационном блоке ожидает на этапе 214 сообщения подтверждения протокола GSMP от аппаратного средства коммутации в базовом коммутационном блоке, указывающего, что отображение осуществлено успешно. Поток при этом коммутируется на уровне 2 полностью аппаратным средством коммутации в базовом коммутационном блоке. Затем базовый коммутационный блок осуществляет пересылку пакета на этапе 176.

На фиг.6c представлена диаграмма, иллюстрирующая этапы, выполняемые при пересылке пакета в системном узле так, как показано для этапа 176 пересылки пакета на фиг.5b. Системный узел на этапе 218 запускает процедуру пересылки пакета. Если поток, к которому принадлежит пакет, не маркирован в нисходящей линии (этап 220), то системный узел передает пакет по виртуальному каналу, установленному по умолчанию в нисходящую линию на этапе 222 и затем переходит к состоянию ожидания 224 для ожидания поступления пакетов. Однако если поток, к которому принадлежит пакет маркирован в нисходящей линии, указывающей, что системный узел ранее принял сообщение "переадресация" протокола IFMP для маркировки этого потока на время жизни, то системный узел на этапе 226 проверяет, не истекло ли время жизни для переадресации данного пакета. Если время жизни не истекло, то системный узел передает пакет по маркированному виртуальному каналу в сообщении "переадресация" протокола IFMP на этапе 228, затем переходит в состояние ожидания 224. Если время жизни истекло, то системный узел автоматически исключает переадресацию потока на этапе 230. Системный узел затем передает пакет по каналу, установленному по умолчанию (этап 222) и возвращается в состояние ожидания на этапе 182, как показано на фиг.5b.

Как описано выше, фиг.6a-6c в общем относятся к взаимодействию системных узлов

(или узлов коммутации) и не затрагивают исполнительные устройства коммутации. Фиг. 6d-6e относятся к взаимодействию узлов коммутации, когда по меньшей мере одно исполнительное устройство коммутации связано с базовым коммутационным блоком, как описано ниже.

На фиг. 6d представлена диаграмма, иллюстрирующая этапы, выполняемые в контроллере коммутации при маркировке потока для пакетов, принимаемых от связанного исполнительного устройства коммутации, являющегося источником, как показано этапом 1660 маркировки потока на фиг.5d. На фиг.6d показаны три сценария: когда поток пакетов желательно передать на другой интерфейс исполнительного устройства коммутации, являющегося источником; когда поток пакетов желательно передать на интерфейс другого связанного исполнительного устройства коммутации, т.е. на исполнительное устройство коммутации, являющееся адресатом; и когда поток пакетов желательно передать на интерфейс другого связанного системного узла (или узла коммутации, такого как другой базовый коммутационный блок, шлюзовой блок коммутации или главный компьютер).

Как показано на фиг.6d, если поток пакетов, принятый от исполнительного устройства коммутации, являющегося источником, желательно передать на другой интерфейс того же самого исполнительного устройства коммутации (как определено на этапе 1662), контроллер коммутации (на этапе 1664) использует протокол IFMP-C для кондиционирования исполнительного устройства коммутации, являющегося источником, чтобы направлять последующие пакеты, принимаемые для данного потока с соответствующим заголовком и преобразования в интерфейсе адресата данного исполнительного устройства коммутации.

Если поток пакетов, принимаемых от исполнительного устройства коммутации, являющегося источником, нежелательно направлять к другому интерфейсу того же самого исполнительного устройства коммутации (как определено на этапе 1662), то на этапе 1666 определяется, не следует ли направлять поток пакетов, принимаемых от исполнительного устройства коммутации, являющегося источником, к интерфейсу исполнительного устройства коммутации, являющегося адресатом. Если это так, то контроллер коммутации (на этапе 1668) выбирает свободную метку x в восходящей линии связи между контроллером коммутации и исполнительным устройством коммутации, являющимся источником, и выбирает (на этапе 1670) свободную метку y в нисходящей линии связи между контроллером коммутации и исполнительным устройством коммутации, являющимся адресатом. Затем контроллер коммутации использует протокол GSMP для отображения x в y на этапе 1672. На этапе 1674 контроллер коммутации использует протокол IFMP-C для кондиционирования исполнительного устройства коммутации, являющегося адресатом, чтобы направлять на интерфейс адресата последующие пакеты, принимаемые с меткой y с соответствующим заголовком и преобразованием. Затем на этапе 1676 контроллер коммутации

использует протокол IFMP-C для кондиционирования исполнительного устройства коммутации, являющегося источником, чтобы направлять последующие пакеты потока с соответствующим заголовком и преобразованием к метке x.

Если поток пакетов, принимаемых от исполнительного устройства коммутации, являющегося источником, нежелательно передавать на интерфейс исполнительного устройства коммутации, являющегося адресатом (что определено на этапе 1666), то поток пакетов, принимаемых от исполнительного устройства коммутации, являющегося источником, желательно передать на интерфейс другого связанного системного узла (или узла коммутации, такого как другой базовый коммутационный блок, шлюзовой блок коммутации или главный компьютер). В этом случае контроллер коммутации (на этапе 1680) выбирает свободную метку x в восходящей линии между контроллером коммутации и исполнительным устройством коммутации, являющимся источником. На этапе 1682 контроллер коммутации ожидает свободной метки y в нисходящей линии, выбираемой коммутационным узлом и пересылаемой посредством протокола IFMP. Затем контроллер коммутации использует протокол GSMP для отображения x в y на этапе 1684. На этапе 1686 контроллер коммутации использует протокол IFMP-C для кондиционирования исполнительного устройства коммутации, являющегося источником, для пересылки последующих пакетов потока с соответствующим заголовком и преобразованием к метке x.

На фиг. 6е представлена диаграмма, иллюстрирующая этапы, выполняемые в контроллере коммутации при маркировке потока (начиная с этапа 1700) для пакетов, которые принимаются от связанного коммутационного узла и предназначены для интерфейса на связанном исполнительном устройстве коммутации. Если поток пакетов, принимаемых от коммутационного узла, являющегося источником, желательно направить к интерфейсу исполнительного устройства коммутации, являющегося адресатом, то контроллера коммутации (на этапе 1702) выбирает свободную метку x в восходящей линии между контроллером коммутации и коммутационным узлом, являющимся источником, и выбирает (на этапе 1704) свободную метку y в нисходящей линии между контроллером коммутации и исполнительным устройством коммутации адресата. Затем контроллер коммутации использует протокол GSMP для отображения x в y на этапе 1706. На этапе 1708 контроллер коммутации использует протокол IFMP-C для приведения исполнительного устройства коммутации адресата в состояние для направления в интерфейс адресата последующих пакетов потока, принимаемого по метке y с соответствующим заголовком и преобразованием. На этапе 1710 контроллер коммутации использует протокол IFMP для запроса коммутационного узла восходящей линии о передаче последующих пакетов потока к метке x. Дополнительные детали приведенного выше описания представлены ниже.

А. Протокол IFMP и передача

маркированного потока по линиям передачи ATM данных

1. Протокол IFMP

Системное программное обеспечение использует протокол управления потоком Ipsilon (IFMP) для обеспечения возможности системному узлу (такому, как базовый коммутационный блок, шлюзовой блок коммутации или главный компьютер/сервер/рабочая станция) классифицировать пакеты протокола IP как принадлежащие к потоку сходных пакетов, основываясь на некоторых общих характеристиках. Потоки определяются "идентификатором потока". Идентификатор потока для конкретного потока дает содержимое или значения набора полей из заголовка пакета, которые определяют поток. Содержимое набора полей из заголовков пакетов является одним и тем же для всех пакетов, принадлежащих данному конкретному потоку. Могут быть определены различные типы пакетов. Каждый тип потока определяет набор полей из заголовка пакета, которые используются для определения потока. Например, один тип потока может определять набор полей из заголовка пакета, который определяет поток как имеющий пакеты, переносящие данные между прикладными задачами, реализуемыми на станциях, в то время как другой тип пакета может определять набор полей из заголовка пакета, который определяет поток как имеющий пакеты, переносящие данные между станциями.

В одном из вариантов настоящего изобретения определены три типа потоков: поток типа 0, поток типа 1 и поток типа 2. Разумеется, могут быть определены и другие или дополнительные типы потоков. Поток типа 0 используется для перехода на формирование пакетов по протоколу IP от формирования пакетов по умолчанию. Формат идентификатора потока для потока типа 0 является нулевым и соответственно имеет нулевую длину. Тип 1 представляет собой тип потока, который определяет набор полей из заголовка пакета, идентифицирующий поток как содержащий пакеты, переносящие данные между прикладными задачами, реализуемыми на станциях. Тип 1 потока полезно использовать для потоков, содержащих пакеты для протоколов, таких, как UDP (протокол дейтаграммы пользователя) и TCP (протокол управления передачей), в которых первые четыре байта после IP-заголовка определяют номер порта источника и номер порта адресата, которые используются для указания исполняемой прикладной задачи. Идентификатор потока для типа 1 потока имеет длину четырех 32-битовых слов. Формат идентификатора потока для типа 1 потока, указанный ссылочной позицией 240 на фиг. 7а, включает (в порядке от старшего бита к младшему биту) версию, длину Интернет-заголовка (IHL), тип обслуживания, время жизни и поля протокола в качестве первого слова; поле адреса источника в качестве второго слова; поле адреса места назначения (адресата) в качестве третьего слова. Эти поля в идентификаторе потока взяты из заголовка IP-пакета потока типа 1. Идентификатор потока для потока типа 1 также включает поля номера порта источника

и номера порта адресата (первые четыре байта в IP-пакете после IP-заголовка) в качестве четвертого слова. Тип 2 представляет собой тип потока, который определяет набор полей из заголовка пакета, идентифицирующий поток как содержащий пакеты, переносящие данные между станциями без определения прикладных задач, выполняемых на станциях. Идентификатор потока для потока типа 2 имеет длину трех 32-битовых слов. Формат идентификатора потока для типа 2 потока, указанный ссылочной позицией 250 на фиг.7b, включает версию, длину Интернет-заголовка (IHL), тип обслуживания, время жизни, протокол, поля адреса источника и адреса места назначения (адресата) из заголовка IP-пакета. Формат идентификатора потока для потока типа 2 тот же самый, что и для потока типа 1, без четвертого слова. Иерархический характер идентификаторов потоков для различных типов потоков позволяет выполнять конкретные операции сравнения для IP- пакетов для облегчения классификации потоков.

Настоящее изобретение обеспечивает возможность обрабатывать различные типы потоков отдельно, в зависимости от типа потока. Потоки, например, такие, как переносящие трафик, содержащие требования к качеству обслуживания или имеющие длительное время удержания, могут быть конфигурированы так, чтобы коммутироваться по мере возможности. Другие типы потоков, такие, например, как потоки короткой длительности или запросы в базы данных, обрабатываются с использованием пересылки пакетов по протоколу IP без установления соединений. Кроме того, каждый тип потока также определяет тип формирования пакета (инкапсуляция), который должен использоваться после того, как этот тип потока будет переадресован. Инкапсуляция для каждого типа потока может быть определена для различных технологий каналов передачи данных. В рассматриваемом варианте осуществления система использует инкапсуляцию для каналов передачи ATM данных, детально описанную ниже.

Конкретный поток пакетов может быть связан с конкретной ATM меткой. В соответствии с рассматриваемым вариантом осуществления метка представляет собой идентификатор виртуального маршрута и идентификатор виртуального канала (VPI/VCI). "Диапазон" меток для конкретного порта представляет собой множество меток (VPI/VCI), предоставленных для использования в этом порте. Предполагается, что виртуальные каналы являются односторонними, так что метка входящего направления каждой линии принадлежит входному порту, с которым она соединена. Разумеется, для вариантов осуществления, использующих другие технологии коммутации, такие как ретрансляция кадров, идентификатор соединения линии передачи данных может использоваться в качестве метки. Для вариантов осуществления с использованием быстродействующей пакетной коммутации в качестве метки может быть использован идентификатор мультиплексирования канала линии передачи

данных.

Как описано выше, классификация потока является решением, принимаемым на местном уровне. Если IP-пакет принимается системным узлом, системный узел передает IP-пакет по каналу, устанавливаемому по умолчанию. Узел также классифицирует IP-пакет как принадлежащий к конкретному потоку, и соответственно принимает решение, надо ли последующие пакеты, принадлежащие тому же самому потоку, коммутировать непосредственно в ATM коммутатор или продолжать пересылать поэтапно с помощью программного обеспечения маршрутизатора в узле. Если принимается решение коммутировать поток пакетов, то узел выбирает для данного потока доступную метку (VPI/VCI) входного порта, на который был принят пакет. Узел, который принял решение коммутировать поток, затем запоминает метку, идентификатор потока и время жизни и передает сообщение "переадресация" (REDIRECT) протокола IFMP в восходящую линию к предыдущему узлу, от которого поступил пакет. Как описано выше, идентификатор потока содержит набор полей заголовка, который характеризует поток. Время жизни определяет длительность времени, в течение которой действует переадресация. Если только состояние потока не обновляется, связь между потоком и меткой должна быть отменена после истечения времени жизни. Истечение времени жизни до момента обновления состояния приводит к тому, что последующие пакеты, принадлежащие к потоку, будут передаваться по каналу передачи, установленному по умолчанию между соседними узлами.

Состояние потока обновляется путем передачи в восходящую линию сообщения "переадресация", имеющего ту же самую метку и идентификатор потока, что и исходные, и имеющего другое время жизни. Сообщение "переадресация" указывает узлу восходящей линии передавать все последующие пакеты, имеющие характеристики, совпадающие с идентифицируемыми идентификатором потока, по виртуальному каналу, определенному меткой. Решение о переадресации представляет собой также решение местного уровня, обрабатываемое узлом восходящей линии, в то время как решение о классификации потока является решением, принимаемым на местном уровне, обрабатываемым узлом нисходящей линии. Соответственно даже если узел нисходящей линии запросит переадресацию конкретного потока пакетов, узел восходящей линии может принять решение принять или игнорировать запрос о переадресации. Кроме того, сообщения "переадресация" не подтверждаются. Вместо этого первый пакет, приходящий по новому виртуальному каналу, служит для указания на то, что запрос переадресации принят.

В настоящем изобретении протокол IFMP системного программного обеспечения включает протокол близости IFMP и протокол переадресации IFMP. Протокол близости IFMP позволяет системному узлу ("хосту", базовому коммутационному блоку или шлюзовому блоку коммутации) определить идентификационные данные системного узла

на другом конце линии связи. Кроме того, протокол близости IFMP используется для синхронизации состояний в линии связи, для определения того, когда системный узел на другом конце линии связи изменяется, и для обмена перечнями IP адресов, присвоенных линии связи. С использованием протокола переадресации IFMP система может направлять сообщения "переадресация" по линии связи только после того, как система использует протокол близости IFMP для идентификации другого системного узла на другом конце линии связи и достижения синхронизации состояний в линии связи. Любое сообщение "переадресация", принимаемое по линии связи, не достигшей синхронизации состояний, должно отбрасываться. Протокол близости IFMP и протокол переадресации IFMP подробно описаны после изложенного ниже подробного описания работы системы.

Для иллюстрации преимуществ, предоставляемых системой, будет рассмотрен конкретный пример, описывающий классификацию потока и переадресацию в соответствии с настоящим изобретением при использовании конфигурации локальной сети, подобной представленной на фиг.2а. В частности, в примере рассматривается взаимодействие между первым и вторым шлюзовыми блоками коммутации 21 и базовым коммутационным блоком 1 по фиг.2а. При запуске системы между программным обеспечением системы, выполняемым на контроллерах базового коммутационного блока 1 и каждого из соседних узлов (в данном примере первого и второго шлюзовых блоков коммутации 21), по умолчанию устанавливается виртуальный канал пересылки ATM данных. Когда IP-пакет передается от основной операционной среды 35₂ локальной сети по линии связи 39₁ сетевого уровня, IP-пакет принимается первым шлюзовым блоком коммутации 21 через одну из его плат сетевого интерфейса локальной сети. Затем программное обеспечение в первом шлюзовом блоке коммутации 21 анализирует IP-пакет и выполняет инкапсуляцию по умолчанию содержимого IP-пакета для передачи по линии 33₄ (установленной между платой ATM сетевого интерфейса первого шлюзового блока коммутации 21 и выбранным портом аппаратных средств ATM коммутатора в базовом коммутационном блоке 1) к базовому коммутационному блоку 1. Затем аппаратные средства ATM коммутатора направляют ATM элементы данных к плате ATM сетевого интерфейса 9 в контроллере коммутации 9, который затем повторно компонует пакет и направляет IP дейтаграмму к системному программному обеспечению в контроллере коммутации для IP маршрутизации. Контроллер коммутации направляет пакет обычным образом по установленному по умолчанию каналу передачи, который был первоначально установлен между базовым коммутационным блоком 1 и вторым шлюзовым блоком коммутации 21 при запуске. Кроме того, контроллер коммутации в базовом коммутационном блоке 1 выполняет классификацию потока по пакету для определения того, следует ли последующие пакеты, принадлежащие тому же самому потоку, непосредственно коммутировать

аппаратными средствами ATM, или продолжать маршрутизировать их поэтапно с использованием системного программного обеспечения. Если программное обеспечение контроллера коммутации принимает решение, что поток должен коммутироваться, то оно выбирает свободную метку (метку x) из пространства меток (пространство меток представляет собой просто определенный диапазон идентификаторов VPI/VCI) входного порта (порта i), на котором принят пакет. Контроллер коммутации также выбирает свободную метку (метку x') на своем порте управления (реальный или виртуальный порт, которым контроллер коммутации соединен с ATM коммутатором). С использованием протокола GSMP системное программное обеспечение подает команду ATM коммутатору отображать метку x на входном порте i на метку x' на порте управления с. После того как коммутатор возвратит назад сообщение подтверждения приема протокола GSMP, контроллер коммутации направляет сообщение "переадресация" (IFMP REDIRECT) в восходящую линию предыдущему узлу (в данном примере первому шлюзовому узлу коммутации 21), от которого поступил данный пакет. Сообщение "переадресация" представляет собой просто команду от базового коммутационного блока 1 к первому шлюзовому узлу коммутации 21 передавать все последующие пакеты с полями заголовка, совпадающими с соответствующими полями, указанными в идентификаторе потока сообщения "переадресация" в виртуальный ATM канал, определенный меткой сообщения "переадресация". Если только состояние потока не будет обновлено, прежде чем истечет время жизни сообщения "переадресация", связь между потоком и меткой сообщения "переадресация" должна быть отменена, что приведет к тому, что последующие пакеты потока будут пересылаться по каналу, установленному по умолчанию (установленному первоначально при запуске) между базовым коммутационным блоком 1 и первым шлюзовым блоком коммутации 21.

Если первый шлюзовой блок коммутации 21 принимает запрос в сообщении "переадресация", пересланном базовым коммутационным блоком 1, пакеты, принадлежащие данному потоку, будут прибывать на порт с контроллера коммутации с меткой x' ATM VPI/VCI. Пакеты будут перекомпоновываться и маршрутизироваться системным программным обеспечением, но процедура ускоряется в результате того, что предыдущее решение о маршрутизации для потока кэшировано и индексировано меткой x' в системном программном обеспечении. Соответственно очевидно, что поток может быть маркирован, но не обязательно должен коммутироваться.

Одно из важных преимуществ коммутации становится очевидным в ситуации, когда узел нисходящей линии (в данном примере второй шлюзовой блок коммутации) также участвует в переадресации того же самого потока. Когда базовый коммутационный блок 1 маршрутизирует первоначальный пакет, принадлежащий потоку, к второму шлюзовому блоку коммутации 21 по установленному по умолчанию между ними каналу, то узел

нисходящей линии (в данном случае второй шлюзовой блок коммутации 21) перекомпоновывает пакет и пересылает его обычным образом. Для пакета, принятого на его порте j, второй шлюзовой блок коммутации 21 также выполняет классификацию потока и принимает решение на своем локальном уровне в виде таблицы, следует ли коммутировать последующие пакеты, принадлежащие этому потоку, или продолжать пересылку пакетов в контроллере. Если второй шлюзовой блок коммутации 21 принимает решение, что последующие пакеты, принадлежащие этому потоку, следует коммутировать, он передает свое собственное сообщение "переадресация" (со свободной меткой у на его порте j, идентификатором потока и временем жизни) в восходящую линию к базовому коммутационному блоку 1. Базовый коммутационный блок 1 может принять или игнорировать данный запрос о переадресации. Если базовый коммутационный блок 1 принимает решение коммутировать поток, то системное программное обеспечение в контроллере коммутации базового коммутационного блока отображает метку х на порте i в метку у на порте j. Таким образом, трафик больше не пересылается процессору управления коммутацией, а непосредственно коммутируется на требуемый выходной порт аппаратных средств ATM коммутатора. Соответственно весь последующий трафик, принадлежащий к данному потоку, может коммутироваться полностью в пределах аппаратных средств ATM коммутатора базового коммутационного блока 1. Если пакет приходит от порта ATM коммутатора базового коммутационного блока 1, то второй шлюзовой блок коммутации 21, использующий свою плату сетевого ATM интерфейса, принимает этот пакет по ATM линии связи 33₅. Затем второй шлюзовой блок коммутации 21 перекомпоновывает и передает пакет через свою плату сетевого интерфейса по линии 39₂ в локальную сеть 35₂. Пользовательское устройство 41, для которого предназначается пакет, получает его из локальной сети 35₂ через плату сетевого интерфейса 43 пользовательского устройства.

Если системный узел (в данном примере базовый коммутационный блок 1) принимает сообщение "переадресация", он также изменяет инкапсуляцию, использованную для переадресуемого потока. Вместо использования установленной по умолчанию инкапсуляции, используемой для IP-пакетов в установленном по умолчанию канале передачи, системный узел может использовать другой тип инкапсуляции в зависимости от типа потока. Таким образом, базовый коммутационный блок 1 инкапсулирует последующие пакеты, принадлежащие данному потоку, и передает их по определенному виртуальному каналу, указанному меткой у. Некоторые типы инкапсуляции могут удалять определенные поля из IP-пакета. Если эти поля удалены, системный узел, который выдал сообщение "переадресация", запоминает поля и связывает поля с определенным виртуальным ATM каналом. В случае рассматриваемого примера, если базовый

коммутационный блок 1 принимает сообщение "переадресация", переданное вторым шлюзовым блоком коммутации 21, то базовый коммутационный блок 1 запоминает поля и связывает поля с определенным виртуальным ATM каналом, определенным меткой у. Аналогично, если первый шлюзовой блок коммутации 21 принимает сообщение "переадресация", пересланное первым коммутационным блоком 1, то первый шлюзовой блок коммутации 21 запоминает поля и связывает поля с определенным виртуальным ATM каналом, определенным меткой х. Полный пакет может быть восстановлен с использованием входящей метки для доступа к запомненным полям. Этот способ обеспечивает определенную степень защищенности путем запрещения пользователю устанавливать коммутируемый поток в разрешенное место назначения или пункт обслуживания за пределами средства защиты и затем изменять заголовок IP-пакета для получения доступа к запрещенному месту назначения.

Каждый системный узел поддерживает таймер фонового восстановления. Если время таймера фонового восстановления истекло, то состояние каждого потока анализируется. Если поток принял трафик с момента последнего периода восстановления, системный узел восстанавливает состояние потока путем направления сообщения "переадресация" в восходящую линию с той же самой меткой и идентификатором потока, что и в исходном сообщении "переадресация", а также с указанием времени жизни. Если поток не принял трафика с момента последнего периода восстановления, то системный узел удаляет кэшированное состояние потока. Системный узел удаляет состояние потока путем выдачи сообщения "восстановление" (RECLAIM) в восходящую линию для восстановления указанной метки для повторного использования. Однако пока узел восходящей линии не передаст сообщение подтверждения приема сообщения "восстановление" (IFMP RECLAIM ACK), которое принимается узлом, передавшим сообщение "восстановление", состояние потока не отменяется и метка не может быть повторно использована. Подтверждение приема сообщения "восстановление" освобождает запрошенную метку. Системный узел определяет, принял ли поток трафик двумя различными путями, в зависимости от того, коммутировался ли поток или нет. Для потоков, которые маркированы, но не коммутировались, контроллер системного узла анализирует свое собственное состояние для проверки того, принял ли поток какой-либо трафик в предыдущий период восстановления. Для потоков, которые коммутировались, контроллер системного узла запрашивает аппаратные средства ATM коммутатора с использованием сообщения протокола GSMP, чтобы проверить, находился ли недавно в активном состоянии конкретный канал. Соответственно в рассматриваемом примере базовый коммутационный блок 1 контролирует трафик, чтобы проверить, был ли отображен данный конкретный поток с первого шлюзового блока коммутации 21 на порт управления базового коммутационного блока 1, или отображен с

первого шлюзового блока коммутации 21 на второй шлюзовой блок коммутации 21 через АТМ коммутатор в базовом коммутационном блоке 1. Если данный поток не принимал недавно трафика в предшествующий период обновления, то базовый коммутационный блок 1 передаст сообщение "восстановление" протокола IFMP и удалит состояние потока, если принято сообщение подтверждения IFMP RECLAIM ACK. Таким образом, второй шлюзовой блок коммутации 21 контролирует трафик, чтобы проверить, был ли отображен данный конкретный поток с порта управления базового коммутационного блока 1 на второй шлюзовой блок коммутации 21. Кроме того, главный компьютер/сервер/рабочая станция, оснащенные соответствующим системным программным обеспечением, также оснащены таймером фонового восстановления. Контролируя трафик, чтобы проверить, был ли какой-либо поток отображен на него, главный компьютер может передать сообщение "восстановление" и удалить состояние потока, если принято сообщение подтверждения IFMP RECLAIM ACK.

Как описано выше, протокол близости IFMP используется для установления синхронизации состояний, а также для идентификации соседних системных узлов и обмена IP-адресами. Для целей протокола близости IFMP системный узел имеет три возможных состояния для конкретной линии связи: SYNSENT (передано сообщение синхронизации), SYNRCVD (принято сообщение синхронизации), ESTAB (синхронизация установлена). Синхронизация состояний в линии связи (когда системный узел достигает состояния ESTAB для линии связи) требуется, прежде чем система сможет передавать какие-либо сообщения переадресации с использованием протокола переадресации IFMP.

На фиг. 8а представлена структура обобщенного сообщения 300 протокола близости IFMP. Все сообщения протокола близости IFMP инкапсулированы в IP-пакете. На фиг. 8b показан обобщенный IP-пакет (в его современной версии IPv4) с полем данных переменной длины, в котором может быть инкапсулировано сообщение протокола близости IFMP. В качестве указателя того, что IP-пакет содержит IFMP-сообщение, поле протокола в IP-заголовке инкапсулированного IP-пакета должно содержать десятичное значение 101. Поле времени жизни в заголовке IP-пакета, инкапсулирующего IFMP-сообщение, устанавливается на 1. Таким образом, все сообщения протокола близости IFMP передаются по ограниченному IP-адресу места назначения передачи (255.255.255.255) с использованием адреса в поле адреса места назначения IP-заголовка. Как показано на фиг. 8а, сообщение 300 протокола близости IFMP включает (в порядке от старшего бита к младшему) следующие поля: 8-битовое поле "версия" (302), 8-битовое поле "Op Code" (304), 16-битовое поле "контрольная сумма" (306) в качестве первого 32-битового слова; экземпляр передатчика (308) в качестве второго 32-битового слова; экземпляр "равного по положению" (310) в качестве третьего 32-битового слова; идентификация "равного по положению" (312) в качестве четвертого 32-битового слова; следующий номер

последовательности "равного по положению" (314) в качестве пятого 32-битового слова и перечень адресов (316), представляющий собой поле переменного числа 32-битовых слов.

В сообщении протокола близости IFMP поле "версия" 302 определяет версию протокола IFMP, используемую в данный момент времени. Поле "Op Code" 304 определяет функцию сообщения протокола близости IFMP. В рассматриваемом варианте осуществления имеются четыре возможных кода, т.е. функции сообщений протокола близости IFMP: SYN (сообщение синхронизации, Op Code = 0), SYNACK (сообщение подтверждения синхронизации, Op Code = 1), RSTACK (сообщение подтверждения сброса, Op Code = 2), ACK (сообщение подтверждения, Op Code = 3). В каждом системном узле необходим таймер для периодического генерирования сообщений SYN, SYNACK, ACK. В рассматриваемом варианте осуществления период таймера равен 1 с, но могут быть определены и другие длительности периодов таймера. Если значение таймера истекло, и системный узел находится в состоянии SYNSENT, то системный узел сбрасывает таймер и передает сообщение SYN протокола близости IFMP. Если значение таймера истекло и системный узел находится в состоянии SYNRCVD, то системный узел сбрасывает таймер и передает сообщение SYNACK протокола близости IFMP. Если значение таймера истекло и системный узел находится в состоянии ESTAB, то системный узел сбрасывает таймер и передает сообщение ACK протокола близости IFMP.

Контрольная сумма 306 представляет собой 16-битовое дополнение до единицы суммы дополнений до единицы следующих величин: поля адреса источника, адреса места назначения, протокола и общей длины сообщения протокола близости IFMP. Контрольная сумма 306 используется системой для целей контроля ошибок.

При обсуждении протокола IFMP понятие "передатчик" относится к системному узлу, который передает сообщение протокола IFMP, а понятие "равный по положению" относится к системному узлу, к которому передатчик передает сообщение протокола IFMP для линии связи.

В сообщениях SYN, SYNACK, ACK протокола близости IFMP экземпляр передатчика 308 представляет собой "номер экземпляра" передатчика для линии связи. Указывая конкретный экземпляр линии, номер экземпляра представляет 32-битовое ненулевое число, для которого гарантирована его уникальность в пределах ближайшего прошлого, и его изменение, если линия или системный узел вводятся в действие после отказа. Соответственно каждая линия имеет свой собственный уникальный номер экземпляра. Экземпляр передатчика используется для определения того, когда линия вновь стала работоспособной после отказа, или когда идентификация равного по положению на другом конце линии изменилась. (Экземпляр передатчика 308 используется аналогично тому, как используется исходный номер последовательности (ISN) в протоколе TCP.) Для сообщения RSTACK протокола близости

IFMP экземпляр передатчика 308 устанавливается на значение поля 310 экземпляра равного по положению из входящего сообщения, которое обусловило формирование сообщения RSTACK.

В сообщениях SYN, SYNACK, ACK протокола близости IFMP поле 310 экземпляра равного по положению представляет собой то, что передатчик считает текущим номером экземпляра для равного по положению для данной линии. Если в передатчике не известен текущий номер экземпляра для равного по положению для данной линии, то поле 310 экземпляра равного по положению устанавливается в нуль. В сообщении RSTACK протокола близости IFMP поле 310 экземпляра равного по положению устанавливается на значение поля 308 экземпляра передатчика из входящего сообщения, которое вызвало генерирование сообщения RSTACK.

Для сообщений SYN, SYNACK, ACK протокола близости IFMP поле 312 идентификации равного по положению представляет собой IP-адрес равного по положению, который, как представляется для данного передатчика сообщения, находится на другом конце линии связи. Передатчик принимает IP-адрес, который находится в поле адреса источника IP-заголовка, инкапсулирующего сообщение SYN или SYNACK, принятое передатчиком, и использует этот IP-адрес в поле 312 идентификации равного по положению в сообщении протокола IFMP, которое он передает. Если передатчик не знает IP-адрес равного по положению на другом конце линии связи, поле 312 идентификации равного по положению устанавливается на нуль. Для сообщения RSTACK протокола близости IFMP поле 312 идентификации равного по положению устанавливается на значение IP-адреса поля адреса источника из IP-заголовка входящего сообщения, которое вызвало генерирование сообщения RSTACK.

Поле 314 следующего номера последовательности равного по положению дает значение поля следующего номера последовательности равного по положению, который передатчик ожидает получить в следующем сообщении протокола переадресации IFMP. Если значение поля 314 следующего номера последовательности равного по положению во входящем сообщении подтверждения (ACK) протокола близости IFMP больше, чем значение, равное единице плюс значение номера последовательности (из последнего сообщения протокола переадресации IFMP, переданного с порта, на который было принято сообщение ACK протокола близости IFMP), то линия должна устанавливаться в исходное состояние.

Поле 316 перечня адресов представляет собой перечень из одного или более IP-адресов, которые присвоены линии передатчиком сообщения протокола близости IFMP. Перечень должен иметь по меньшей мере одну запись, которая идентична адресу источника IP-заголовка сообщения протокола близости IFMP. Содержимое перечня не используется протоколом IFMP, а может предоставляться для протокола маршрутизации.

На фиг.8с представлена диаграмма,

иллюстрирующая работу системного узла после приема пакета с входящим сообщением протокола близости IFMP. После запуска системы системный узел принимает пакет с входящим сообщением протокола близости IFMP (этап 320). На этапе 322 системный узел определяет, является ли входящее сообщение протокола близости IFMP сообщением RSTACK. Если входящее сообщение протокола близости IFMP не является сообщением RSTACK (например, сообщение SYN, SYNACK, ACK), то системный узел работает в соответствии с тем, как показано на диаграмме состояний на фиг.8d. Если входящее сообщение протокола близости IFMP является сообщением RSTACK, то системный узел проверяет на этапе 324, совпадают ли значения экземпляра передатчика и IP-адреса источника во входящем сообщении RSTACK со значениями, запомненными из предыдущего сообщения посредством операции обновления верификатора равного по положению. Для протокола близости IFMP операция обновления верификатора равного по положению определяется как запоминание экземпляра передатчика и IP-адреса источника из сообщения SYN или SYNACK, принятого от равного по положению на конкретный порт. Если на этапе 324 установлено, что указанные значения совпадают, то системный узел на этапе 326, определяет совпадают ли значения экземпляра равного по положению и идентификации равного по положению во входящем сообщении RSTACK со значениями экземпляра передатчика и IP-адреса источника, используемыми для всех сообщений SYN, SYNACK или ACK, передаваемых с порта, на который было принято входящее сообщение RSTACK. Если на этапе 326 установлено, что указанные значения совпадают, то системный узел на этапе 328 определяет, находится ли системный узел в состоянии SYNSENT. Если системный узел не находится в состоянии SYNSENT, то системный узел осуществляет установку линии в исходное состояние на этапе 30. Если на этапе 324 или на этапе 326 установлено, что значения не совпадают, или системный узел не находится в состоянии SYNSENT, то системный узел на этапе 332 отбрасывает входящее сообщение RSTACK и ожидает прихода другого пакета. Соответственно, когда в системный узел приходит сообщение RSTACK протокола близости IFMP, системный узел устанавливает линию в исходное состояние, как показано этапами 334, 336, 338, 340 и 342. На этапе 344 системный узел генерирует новый номер экземпляра для линии. Затем системный узел на этапе 336 отменяет верификатор равного по положению (т.е. устанавливает запомненные значения экземпляра передатчика и IP-адреса источника для равного по положению в нуль). На этапе 338 системный узел устанавливает номер последовательности и следующий номер последовательности для равного по положению в нуль. Затем системный узел передает сообщение SYN протокола близости IFMP на этапе 340 и на этапе 342 входит в состояние SYNSENT. Системный узел затем принимает другой пакет для его обработки.

На фиг.8d представлена диаграмма

состояний, иллюстрирующая работу передающего системного узла, когда входящее сообщение протокола близости IFMP не является сообщением RSTACK. Для последующего описания фиг.8d условие "%B" определяется следующим образом: экземпляр передатчика и IP-адрес источника во входящем сообщении совпадают со значениями, запомненными из предыдущего сообщения посредством операции обновления верификатора равного по положению для порта, на который было принято входящее сообщение протокола близости IFMP. Условие "%C" на фиг.8d определяется следующим образом: экземпляр равного по положению и идентификация равного по положению во входящем сообщении совпадают со значениями экземпляра передатчика и IP-адреса источника, используемыми в настоящее время для всех сообщений SYN, SYNACK, ACK, переданных с порта, на котором было принято входящее сообщение протокола близости IFMP. На фиг.8d условие "A" означает, что передающий системный узел принимает входящее сообщение SYNACK протокола близости IFMP, и что условие "%C" удовлетворено. Условие "B" означает, что передающий системный узел принимает входящее сообщение SYNACK протокола близости IFMP, и что условие "%C" не удовлетворено. Условие "C" означает, что передающий системный узел принимает входящее сообщение ACK протокола близости IFMP, и что условия "%B" и "%C" удовлетворены. И условие "D" означает, что передающий системный узел принимает входящее сообщение ACK протокола близости IFMP, и что условия "%B" и "%C" не удовлетворены.

Если передатчик находится в состоянии 350 SYNSENT и принимает входящее сообщение SYN протокола близости IFMP от равного по положению на другом конце линии связи, то передатчик выполняет операцию обновления верификатора равного по положению и передает сообщение SYNACK протокола близости IFMP к равному по положению (показано как этап 352). Затем передатчик переходит из состояния 350 SYNSENT в состояние 354 SYNRCVD. Если передатчик принимает входящее сообщение SYN протокола близости IFMP, находясь в состоянии 354 SYNRCVD, то на этапе 352 передатчик выполняет операцию обновления верификатора равного по положению и передает сообщение SYNACK протокола близости IFMP к равному по положению, но остается в состоянии 354 SYNACK. Если передатчик находится в состоянии 354 SYNRCVD и либо состояние B, либо состояние D удовлетворено, то передатчик передает сообщение RSTACK протокола близости IFMP к равному по положению (показано как этап 356) и остается в состоянии 354 SYNRCVD. Если передатчик находится в состоянии 354 SYNRCVD и состояние C удовлетворено, то передатчик передает сообщение ACK протокола близости IFMP к равному по положению (показано как этап 358) и переходит в состояние 360 ESTAB. Если передатчик находится в состоянии 354 SYNRCVD и состояние A удовлетворено, то передатчик выполняет операцию идентификатора обновления

верификатора равного по положению, передает сообщение ACK протокола близости IFMP к равному по положению (показано как этап 362) и переходит в состояние 360 ESTAB. Передатчик продолжает находиться в состоянии 360 ESTAB, если передатчик принимает сообщение SYN или SYNACK протокола близости IFMP, или если удовлетворено условие C. Если условие D удовлетворено, когда передатчик находится в состоянии 360 ESTAB, то передатчик остается в состоянии 360 ESTAB и передает сообщение RSTACK протокола близости IFMP (показано как этап 356). В состоянии SYNSENT, если либо передатчик принимает сообщение ACK протокола близости IFMP, либо удовлетворено условие B, то передатчик остается в состоянии 350 SYNSENT и передает сообщение RSTACK протокола близости IFMP (этап 356).

Если условие A удовлетворено, когда передатчик находится в состоянии 350 SYNSENT, то передатчик выполняет операцию обновления верификатора равного по положению и передает сообщение ACK протокола близости IFMP (этап 362) и переходит в состояние 360 ESTAB.

Как описано выше, протокол переадресации IFMP используется для передачи сообщений переадресации по линии связи, после того как система использовала протокол близости IFMP для идентификации другого системного узла на другом конце линии связи для достижения синхронизации состояний в линии связи. Любое сообщение переадресации протокола IFMP, принятое в линии, которая еще не осуществила синхронизации состояний, должно быть отвергнуто. На фиг.9a представлена обобщенная структура сообщения 380 протокола переадресации IFMP. Подобно всем сообщениям протокола близости IFMP, все сообщения протокола переадресации IFMP инкапсулированы в IP-пакете. На фиг.8b показан обобщенный IP-пакет (в его современной версии IPv4) с переменной длиной поля данных, в котором может быть инкапсулировано сообщение протокола переадресации. В качестве указания на то, что IP-пакет содержит сообщение протокола IFMP, поле протокола IP-заголовка инкапсулирующего IP-пакета должно содержать десятичное значение 101, а поле "время жизни" в заголовке IP-пакета, инкапсулирующего сообщение протокола IFMP, установлено в 1. Сообщение протокола переадресации IFMP передается в IP-адрес равного по положению на другом конце линии (IP-адрес может быть получен из протокола близости IFMP) с использованием IP-адреса в поле адреса места назначения IP-заголовка. Как показано на фиг.9a, сообщение 380 протокола переадресации IFMP включает (в порядке от старшего бита к младшему): 8-битовое поле "версия" (382), 8-битовое поле "Op Code" (384), 16-битовое поле "контрольная сумма" (386) в качестве первого 32-битового слова; экземпляр передатчика (388) в качестве второго 32-битового слова; экземпляр "равного по положению" (390) в качестве третьего 32-битового слова; номер последовательности (392) в качестве четвертого 32-битового слова и тело сообщения (394), представляющее собой поле переменной длины из 32-битовых слов.

В сообщении протокола переадресации IFMP поле "версия" 382 определяет версию протокола IFMP, используемую в данный момент времени. Поле "Op Code" 384 определяет функцию сообщения протокола переадресации IFMP. В рассматриваемом варианте осуществления имеются пять возможных кодов, т.е. функции сообщений протокола переадресации IFMP: REDIRECT (сообщение переадресации потока, Op Code=4), RECLAIM (сообщение восстановления метки, Op Code=5), RECLAIM ACK (сообщение подтверждения восстановления метки, Op Code=6), LABEL RANGE (сообщение диапазона меток, Op Code=7) и ERROR (сообщение ошибки, Op Code=8).

Контрольная сумма 386 представляет собой 16-битовое дополнение до единицы суммы дополнений до единицы следующих величин: полей адреса источника, адреса места назначения, протокола из IP-пакета, инкапсулирующего сообщение протокола переадресации IFMP, и общей длины сообщения протокола переадресации IFMP. Контрольная сумма 386 используется системой для целей контроля ошибок.

В сообщении протокола переадресации IFMP экземпляр передатчика 388 представляет собой номер экземпляра передатчика для линии, полученный из протокола близости IFMP. В сообщениях протокола переадресации IFMP поле 390 экземпляра равного по положению представляет то, что, как представляется в передатчике, является текущим номером экземпляра равного по положению для линии связи, как это определено из протокола близости IFMP.

Поле 392 номера последовательности позволяет системному узлу, принимающему сообщение протокола переадресации IFMP, обрабатывать сообщения протокола переадресации IFMP по порядку. Номер последовательности 392 получает приращение на единицу, по модулю 2^{32} , на каждое сообщение протокола переадресации IFMP, передаваемое по линии связи. Протокол близости IFMP устанавливает номер последовательности в нуль при установке линии связи в исходное состояние.

Поле 316 тела сообщения содержит перечень одного или более элементов сообщения протокола переадресации IFMP. Все элементы сообщения в перечне имеют один и тот же тип сообщения, так как поле 384 "Op Code" применимо ко всему сообщению протокола переадресации IFMP. Число элементов сообщения, включенных в один пакет, не должно приводить к тому, что суммарный размер сообщения протокола переадресации IFMP превысит максимальный размер передаваемого блока (MTU) для данной линии передачи данных. Для сообщений "диапазон меток" и "ошибка" протокола переадресации IFMP используется один элемент сообщения.

На фиг.9b представлена обобщенная диаграмма, иллюстрирующая работу системного узла после приема сообщения протокола переадресации IFMP. После запуска системный узел на этапе 400 принимает пакет, инкапсулирующий сообщение протокола переадресации IFMP. На этапе 402 системный узел проверяет, достигнута ли протоколом близости IFMP

синхронизация состояний для линии связи. Если синхронизация состояний не достигнута, системный узел отбрасывает пакет, инкапсулирующий принятое сообщение протокола переадресации IFMP (показано этапом 404). Если синхронизация состояний достигнута, то системный узел на этапе 406 проверяет IP-адрес источника, экземпляр передатчика 388 и экземпляр равного по положению 390 для пакета сообщения протокола переадресации IFMP. Если системный узел на этапе 408 определит, что поля экземпляра передатчика 388 и IP-адреса источника входящего сообщения протокола переадресации IFMP не совпадают со значениями, запомненными в операции обновления верификатора равного по положению протокола близости IFMP для порта, на котором принято входящее сообщение протокола переадресации IFMP, то системный узел отбрасывает пакет входящего сообщения протокола переадресации IFMP (этап 404). Если на этапе 408 обнаружено, что указанные значения совпадают, то системный узел на этапе 410 определяет, совпадает ли поле 390 экземпляра равного по положению с текущим значением для экземпляра передатчика протокола близости IFMP. Если на этапе 408 определено, что указанные значения не совпадают, то системный узел отбрасывает пакет (этап 404). Однако если на этапе 408 определено, что эти значения совпадают, то системный узел продолжает (412) обрабатывать принятое сообщение протокола переадресации IFMP в соответствии с тем, как это необходимо.

Как описано выше, сообщение протокола переадресации IFMP может представлять собой сообщение REDIRECT переадресации, которое используется для подачи команды соседнему узлу присвоить одну или более меток пакетам, принадлежащим одному или нескольким конкретным потокам, причем для каждого в течение определенного времени. Системный узел, принявший сообщение REDIRECT, от узла нисходящей линии, принимает решение, следует ли принять запрос переадресации, полученный в сообщении REDIRECT, и осуществлять переадресацию потока. Сообщение REDIRECT не подтверждается обычным способом. Вместо этого реальная переадресация пакетов с присвоенными метками для конкретных потоков указывает, что системный узел принял запрос переадресации, содержащийся в сообщении REDIRECT. Каждый элемент сообщения REDIRECT в теле сообщения 394 сообщения REDIRECT имеет структуру, показанную на фиг.9c. Элемент 420 сообщения REDIRECT включает в себя (в порядке от старшего бита к младшему) следующие поля: 8-битовое поле 422 типа потока, 8-битовое поле 424 длины идентификатора потока, 16-битовое поле 426 времени жизни в составе первого 32-битового слова; 32-битовое поле 428 метки в качестве второго 32-битового слова; идентификатор потока 430, который представляет собой поле из целого кратного 32-битовых слов. Поле 422 типа потока определяет тип потока идентификатора потока, содержащегося в поле 430 идентификатора потока, а поле 424 длины идентификатора потока определяет длину

поля 430 идентификатора потока в целых кратных 32-битового слова. Поле 425 времени жизни определяет длительность времени (в секундах), в течение которой действует переадресация. Как описано в общем виде выше, после истечения интервала времени, определенного в поле 426 времени жизни, связь между идентификатором потока и меткой отменяется. Поле 428 метки содержит метку для конкретного потока, формат которой зависит от типа физической линии связи, по которой пересылается сообщение протокола переадресации IFMP. Поле 430 идентификатора потока идентифицирует поток, с которым связана конкретная метка в поле 428 метки.

В элементах сообщения протокола переадресации IFMP тип 0 потока имеет поле "тип потока"= 0 и поле "длина идентификатора потока"=0; тип 1 потока имеет поле "тип потока"= 1 и поле "длина идентификатора потока"=4; тип 2 потока имеет поле "тип потока"=2 и поле "длина идентификатора потока"=3.

Общая обработка сообщений переадресации узлами передатчика и равного по положению в линии связи будет детально описана выше. Кроме того, другие особенности элементов сообщения REDIRECT включают управление метками и контроль ошибок. Если метка в поле 428 меток элемента 420 сообщения REDIRECT находится вне диапазона, который может обрабатываться в соответствующей линии связи, то сообщение LABEL RANGE ("диапазон метки") может быть направлено к передатчику элемента сообщения REDIRECT. Это сообщение информирует передатчик о диапазоне меток, который может быть использован при передаче по линии связи. Если системный узел принимает элемент сообщения REDIRECT, определяющий поток как уже переадресованный, то системный узел проверяет поле метки в принятом сообщении REDIRECT в сравнении с меткой, запомненной для переадресованного потока. Если метки совпадают, то системный узел сбрасывает время жизни переадресованного потока на значение, содержащееся в поле "время жизни" 426 принятого элемента сообщения REDIRECT. Если метки не совпадают, то системный узел игнорирует принятый элемент сообщения, и поток возвращается в состояние, установленное по умолчанию. Если системный узел обнаруживает какую-либо ошибку в любом из полей элемента сообщения REDIRECT, то данный конкретный элемент сообщения REDIRECT отбрасывается. Однако любые другие не содержащие ошибок элементы сообщения REDIRECT, которые могут быть в том же самом теле сообщения переадресации протокола IFMP, не отбрасываются и не подвергаются изменениям каким-либо образом. Системный узел отвечает сообщением ERROR соседнему узлу, который передал элемент сообщения REDIRECT, если для данного системного узла не понятна версия протокола IFMP в принятом сообщении протокола IFMP. Таким образом, если для системного узла не понятен тип потока в каком-либо их элементов сообщения REDIRECT в принятом сообщении протокола IFMP, то системный узел передает сообщение ERROR для каждого типа потока, который ему

не понятен, соседнему узлу, который передал указанный конкретный элемент сообщения REDIRECT.

Как пояснено выше, сообщение протокола переадресации IFMP может быть сообщением RECLAIM (восстановление), которое используется для указания соседнему узлу на необходимость освобождения одного или нескольких потоков от связанных с ними меток, с которыми они в данный момент связаны, и освободить метки для повторного их использования. Системный узел, принимающий элемент сообщения RECLAIM от узла в нисходящей линии, освобождает метку и передает узлу в нисходящей линии элемент сообщения RECLAIM ACK (подтверждение восстановления) в качестве формального подтверждения приема сообщения RECLAIM. Каждый элемент сообщения RECLAIM в теле сообщения 394 имеет структуру, показанную на фиг.9d. Элемент 432 сообщения RECLAIM (в порядке от старшего бита к младшему) содержит следующие поля: 8-битовое поле 434 типа потока, 8-битовое поле 436 длины идентификатора потока и 16-битовое резервное поле 438 в составе первого 32-битового слова; 32-битовое поле 440 метки в качестве второго 32-битового слова; и идентификатор потока 442, представляющий собой поле в целое кратное 32-битового слова. Поле 434 типа потока определяет тип потока идентификатора потока, содержащегося в поле 442 идентификатора потока, а поле 436 длины идентификатора потока определяет длину поля 442 идентификатора потока в целых кратных 32-битового слова. В рассматриваемом варианте осуществления резервное поле 438 не используется и установлено на нуль системным узлом, передающим элемент сообщения RECLAIM, и игнорируется системным узлом, принимающим элемент сообщения RECLAIM. Поле 440 метки содержит метку, которую следует освободить. Поле 442 идентификатора потока идентифицирует поток, относительно которого определенная метка, указанная в поле 440 метки, должна быть освобождена. Каждый элемент сообщения RECLAIM применяется к одному потоку и одной метке. После того как системный узел примет элемент сообщения RECLAIM, удалит из потока метку, возвратит поток в состояние пересылки, устанавливаемое по умолчанию, и освободит метку, системный узел должен сформировать элемент сообщения RECLAIM ACK (подтверждение восстановления). Элементы сообщения RECLAIM ACK могут группироваться вместе в одно или несколько сообщений RECLAIM ACK и возвращаться к передатчику в качестве подтверждения завершения операции восстановления.

Дополнительными особенностями элемента сообщения RECLAIM являются управление меткой и контроль ошибок. Если системный узел принял элемент сообщения RECLAIM, определяющий неизвестный поток, то системный узел возвращает элемент сообщения RECLAIM ACK с теми же полями метки 440 и идентификатора потока 442 к передатчику элемента сообщения RECLAIM. Если системный узел принял элемент сообщения RECLAIM, который указывает на известный поток с меткой в поле 440 метки,

которая в данный момент не связана с данным потоком, то системный узел отсоединяет данный поток от метки и возвращает поток в состояние пересылки, установленное по умолчанию, а также формирует элемент сообщения RECLAIM ACK, содержащий действительную метку, с которой поток был ранее связан, передатчику элемента сообщения RECLAIM. Если системный узел обнаруживает ошибку в каком-либо из полей элемента сообщения RECLAIM, данный конкретный элемент сообщения RECLAIM, содержащий ошибку, отбрасывается. Однако любые другие, не содержащие ошибок элементы сообщения RECLAIM, которые могут быть в том же самом теле сообщения восстановления протокола IFMP, не отбрасываются и не подвергаются изменению каким-либо образом. Системный узел отвечает сообщением ERROR соседнему узлу, который передал элемент сообщения RECLAIM, если для данного системного узла не понятна версия протокола IFMP в принятом сообщении протокола IFMP. Таким образом, если для системного узла не понятен тип потока в каком-либо из элементов сообщения RECLAIM в принятом сообщении протокола IFMP, то системный узел передает сообщение ERROR для каждого типа потока, который ему не понятен, соседнему узлу, который передал указанный конкретный элемент сообщения RECLAIM.

Как упомянуто выше, сообщение протокола переадресации IFMP может представлять собой сообщение RECLAIM ACK, которое используется для подтверждения успешного освобождения одной или нескольких освобождаемых меток. После того как системный узел, принявший элемент сообщения RECLAIM от узла в нисходящей линии, освободит метку, элемент сообщения RECLAIM ACK передается к узлу, который передал элемент сообщения RECLAIM. Если возможно, то каждый элемент сообщения RECLAIM ACK не следует передавать, пока не будут переданы все данные, находящиеся в очереди для передачи по линии связи, с использованием метки, определенной для освобождения. Каждый элемент сообщения RECLAIM ACK в теле сообщения 394 имеет структуру, показанную на фиг.9е. Элемент 444 сообщения RECLAIM ACK (в порядке от старшего бита к младшему) содержит следующие поля: 8-битовое поле 446 типа потока, 8-битовое поле 448 длины идентификатора потока и 16-битовое резервное поле 450 в составе первого 32-битового слова; 32-битовое поле 452 метки в качестве второго 32-битового слова; и идентификатор потока 454, представляющий собой поле в целое кратное 32-битового слова. Поле 446 типа потока определяет тип потока идентификатора потока, содержащегося в поле 454 идентификатора потока, а поле 448 длины идентификатора потока определяет длину поля 454 идентификатора потока в целых кратных 32-битового слова. В рассматриваемом варианте осуществления резервное поле 450 не используется и установлено на нуль системным узлом, передающим элемент сообщения RECLAIM ACK, и игнорируется системным узлом, принимающим элемент сообщения RECLAIM ACK. Поле 452 метки

содержит метку, освобожденную от потока, определенного полем 454 идентификатора потока. Поле 454 идентификатора потока содержит идентификатор потока из элемента сообщения RECLAIM, в котором было запрошено освобождение метки, указанной в поле 452 метки.

Дополнительными особенностями элемента сообщения RECLAIM являются управление меткой и контроль ошибок. Если системный узел принял элемент сообщения RECLAIM ACK, определяющий поток, для которого не выдавалось сообщение RECLAIM, то элемент сообщения RECLAIM ACK игнорируется. Если системный узел принял элемент сообщения RECLAIM ACK, который указывает на другую метку, отличающуюся от метки, переданной в сообщении RECLAIM для данного потока, то системный узел обрабатывает принятый элемент сообщения RECLAIM ACK, как если бы операция восстановления для метки, переданной в сообщении RECLAIM, была успешной. Если системный узел обнаруживает ошибку в каком-либо из полей элемента сообщения RECLAIM ACK, данный конкретный элемент сообщения RECLAIM ACK, содержащий ошибку, отбрасывается. Однако любые другие, не содержащие ошибок элементы сообщения RECLAIM ACK, которые могут быть в том же самом теле сообщения подтверждения восстановления протокола IFMP, не отбрасываются и не подвергаются изменению каким-либо образом. Системный узел отвечает сообщением ERROR соседнему узлу, который передал элемент сообщения RECLAIM ACK с ошибкой, если для данного системного узла не понятна версия протокола IFMP в принятом сообщении протокола IFMP. Таким образом, если для системного узла не понятен тип потока в каком-либо из элементов сообщения RECLAIM ACK в принятом сообщении протокола IFMP, то системный узел передает сообщение ERROR для каждого типа потока, который ему не понятен, соседнему узлу, который передал указанный конкретный элемент сообщения RECLAIM ACK.

Как описано выше, сообщение протокола переадресации IFMP может представлять собой сообщение LABEL RANGE (диапазон меток), которое используется в ответ на сообщение REDIRECT (переадресация), если запрошенная метка в одном или нескольких элементах сообщения REDIRECT находится вне диапазона, который может быть обработан системным узлом, принимающим сообщение переадресации. Сообщение LABEL RANGE информирует передатчик сообщения REDIRECT о диапазоне меток, который может обрабатываться в данной линии связи. В сообщении LABEL RANGE используется один элемент сообщения диапазона меток. Элемент сообщения LABEL RANGE в теле сообщения 394 имеет структуру, представленную на фиг.9f. Элемент сообщения 456 LABEL RANGE содержит поле 458 минимальной метки в виде первого 32-битового слова и поле 460 максимальной метки в виде второго 32-битового слова. Поле 458 минимальной метки и поле 460 максимальной метки соответственно представляют собой минимальное и максимальное значение метки, которая может быть определена в

сообщении протокола переадресации IFMP в конкретной линии связи. Только эти значения меток в пределах диапазона от минимальной метки до максимальной метки (включительно) могут быть определены в сообщении протокола переадресации IFMP в линии связи.

Как описано выше, сообщение протокола переадресации IFMP может представлять собой сообщение ERROR (ошибка), которое может передаваться в ответ на любое сообщение протокола переадресации IFMP. В сообщении ERROR используется один элемент сообщения ошибки. Элемент сообщения ERROR в теле сообщения 394 имеет структуру, представленную на фиг.9g. Элемент 462 сообщения ERROR (в порядке от старшего бита к младшему) содержит 8-битовое поле 464 кода ошибки и 24-битовое поле 466 параметра в виде 32-битового слова. Поле 464 кода ошибки определяет, какой тип ошибки имел место. Сообщение ERROR может определять один параметр. Если системный узел обнаруживает ошибку в любом из полей в элементе сообщения протокола переадресации IFMP, то данный конкретный элемент сообщения с ошибкой отбрасывается и выдается сообщение ERROR. Если системный узел не может осуществить обработку или ему не понятна конкретная версия протокола IFMP в принятом сообщении протокола IFMP, то системный узел передает сообщение ERROR с полем кода ошибки 464, установленным в 1, и с полем 466 параметра, обеспечивающим самую последнюю версию протокола IFMP, которую передатчик может понять или осуществить соответствующую обработку. Кроме того, если системному узлу не понятен тип потока в каком-либо из элементов сообщения протокола переадресации IFMP, который вызвал ошибку, то системный узел передает сообщение ERROR с полем 464 кода ошибки, установленным на 2, и с полем 466 параметра, указывающим тип потока, который вызвал ошибку.

2. Передача маркированного потока по линиям передачи ATM данных

Настоящее изобретение использует линии передачи ATM данных для передачи IP-пакетов между системными узлами. Пакеты, передаваемые по линиям передачи ATM данных, представляют собой потоки, маркированные и инкапсулированные различным образом, в зависимости от типа потока, как указано выше. С использованием классификации потоков настоящее изобретение обеспечивает осуществление обработки разных типов пакетов различным образом (маршрутизация на уровне 2 или коммутация на уровне 3), в зависимости от типа пакета. Кроме того, каждый тип пакета также определяет инкапсуляцию, которая должна использоваться после переадресации данного типа пакета. В рассматриваемом варианте осуществления изобретения система использует инкапсуляции для линий передач ATM данных, как описано подробно ниже. Разумеется, инкапсуляции для каждого типа потока могут быть определены для различных технологий линий передачи данных, для различных аппаратных средств коммутации, которые могут использоваться в соответствии с настоящим изобретением.

Конкретный поток пакетов может быть

связан с конкретной меткой асинхронного режима передачи (ATM меткой). На фиг.10a показан формат 32-битового поля для ATM меток в рассматриваемой системе. Как описано выше, метка представляет собой идентификатор виртуального маршрута и идентификатор виртуального канала (VPI/VCI), причем предполагаются односторонние виртуальные каналы. В порядке от старшего бита к младшему поле 470 ATM метки, показанное на фиг.10a, содержит 4-битовое резервное поле 472, 12-битовое поле 474 VPI и 16-битовое поле 476 VCI. В рассматриваемом варианте осуществления резервное поле 472 установлено на нуль передающим сетевым узлом и игнорируется системным узлом, принимающим ATM метку. Для линии передачи данных, которая не поддерживает полный 12-битовый VPI, неиспользованные биты в поле 474 VPI представляют собой старшие биты поля 474, установленные на нуль. Аналогично для линии передачи данных, которая не поддерживает полный 16-битовый VCI, неиспользованные биты в поле 476 VCI представляют собой старшие биты поля 476, установленные на нуль.

Для любых пакетов в потоке, который не переадресован, системный узел использует инкапсуляцию IP-пакетов, установленную по умолчанию. Если системный узел принимает решение, что конкретный тип пакета должен быть переадресован, то системный узел использует инкапсуляцию, выбранную конкретно для данного типа потока. После приема потока системный узел изменяет инкапсуляцию, использованную для переадресуемого потока, с установленной по умолчанию на иную. Вместо использования для IP-пакетов инкапсуляции, установленной по умолчанию в канале пересылки, установленном по умолчанию, системный узел использует другой тип инкапсуляции в зависимости от типа потока, который переадресуется. Следует иметь в виду, что инкапсулированный IP-пакет режима ATM может представлять собой IP-пакет, который сам инкапсулирует сообщение протокола IFMP, передаваемое к (и/или от) главному компьютеру/серверу/рабочей станции, на которых выполняется комплект системного программного обеспечения, базовому коммутационному блоку или шлюзовому блоку коммутации.

Как описано выше, в возможном варианте осуществления изобретения определены три типа потоков: тип 0 потока, тип 1 потока, тип 2 потока. Тип 0 потока используется для изменения инкапсуляции IP-пакетов с установленной по умолчанию. Тип 1 потока используется для пакетов, переносящих данные между прикладными программами, исполняемыми на станциях. Тип 2 потока используется для пакетов, переносящих данные между станциями, без идентификации прикладных задач, которые могут выполняться на станциях.

В настоящем изобретении инкапсуляция, устанавливаемая по умолчанию, для IP-пакетов в линии передачи ATM данных представляет собой инкапсуляцию, описываемую процедурой. Логическое управление линией связи/точка подсетевого соединения (LLC/SNAP), как показано на фиг.10b. Фиг.10b иллюстрирует IP-пакет,

инкапсулированный по умолчанию. В основном инкапсуляция по умолчанию предусматривает снабжение IP-пакета заголовком LLC/SNAP, причем IP-пакет инкапсулируется в составе полезной нагрузки блока данных протокола субслоя сходимости на уровне ATM адаптации типа 5 (AAL-5 CPCS-PDU). Инкапсулированный по умолчанию IP-пакет 480 содержит заголовок LLC/SNAP (24-битовое поле 482 LLC, за которым следует 8-битовая часть заголовка 484 SNAP в первом 32-битовом слове, и оставшаяся 32-битовая часть заголовка 484 SNAP), IP-пакет 486 (который имеет длину целого кратного 32-битового слова), поле 488 заполнителя и концевое поле 490 AAL-5 CPCS-PDU. Поле заполнителя 488 может иметь длину от 0 до 47 байтов, поле концевика 490 равно 8 байтам (4 32-битовых слова). Блок передачи IP-пакета 486, использующего инкапсуляцию по умолчанию, равен 1500 байтов. Пакеты, использующие инкапсуляцию по умолчанию, передаются к предварительно определенному VPI/VCI (VPI = 0, VCI = 15), который представляет собой установленные по умолчанию идентификаторы виртуального маршрута и виртуального канала для соответствующей аппаратуры (т.е. пакеты передаются по виртуальному каналу, установленному по умолчанию).

Тип 0 потока используется для изменения инкапсуляции IP-пакетов с установленной по умолчанию. На фиг.10с показан инкапсулированный IP-пакет 492 типа 0. IP-пакеты, использующие тип 0 потока, инкапсулируются непосредственно в полезной нагрузке AAL-5 CPCS-PDU, без приписывания заголовка LLC/SNAP. Инкапсулированный IP-пакет 492 типа 0 включает в себя IP-пакет 494 (имеющий длину, равную целому кратному 32-битового слова), поле заполнителя 498 и поле хвостовика 498 AAL-5 CPCS-PDU. Поле заполнителя 496 может занимать от 0 до 47 байтов, а поле хвостовика 498 равно 8 байтам (четыре 32-битовых слова). Блок передачи IP-пакета 494, использующего инкапсуляцию типа 0 потока, равен 1500 байтов. Пакеты, принадлежащие к потоку, переадресуемому из установленного по умолчанию виртуального канала, используют инкапсуляцию типа 0 потока и передаются в VPI/VCI, определенные в поле метки элемента сообщения REDIRECT (переадресация) протокола IFMP, инкапсулированного в IP-пакете 494 (элемент сообщения REDIRECT, инкапсулированный в IP-пакете 494, передается с инкапсуляцией типа 0 потока).

Инкапсуляции, установленные по умолчанию и связанные с типом 0 потока, не предусматривают удаление каких-либо полей из инкапсулируемого IP-пакета. Однако инкапсуляции в соответствии с типом 1 потока и типом 2 потока связаны с удалением определенных полей из IP-пакета. Если эти поля удаляются, то системный узел, который выдал сообщение REDIRECT, запоминает удаленные поля и связывает эти поля с виртуальным ATM каналом, определенным в ATM метке. Соответственно, полный IP-пакет может быть восстановлен в месте назначения с использованием входящей ATM метки для доступа к запомненным полям.

Тип 1 потока используется для пакетов, переносящих данные между прикладными задачами, выполняемыми на станциях. На фиг.10d иллюстрируется IP-пакет, инкапсулированный в соответствии с типом 1 потока. IP-пакеты, использующие инкапсуляцию типа 1 потока, по существу декомпонуются и выбранные части декомпонованных IP-пакетов инкапсулируются непосредственно в полезную нагрузку AAL-5 CPCS-PDU, без приписывания заголовка LLC/SNAP. Инкапсулированный согласно типу 1 потока IP-пакет 500 содержит 16-битовое поле 502 полной длины и 16-битовое поле 504 идентификации из IP-заголовка декомпонованного IP-пакета в качестве первого 32-битового слова. Значение поля 502 полной длины не изменяется, а сохраняет полную длину IP-пакета до декомпоновки. Инкапсулированный согласно типу 1 потока IP-пакет 500 также содержит 8-битовое поле 506 флага, 12-битовое поле 508 смещения фрагмента и 16-битовое поле 510 контрольной суммы из IP-заголовка декомпонованного IP-пакета в качестве второго 32-битового слова. Передаваемое значение поля 510 контрольной суммы представляет собой значение контрольной суммы, которое было бы вычислено для полного IP-заголовка, если бы поле TTL было установлено в нуль. Поля "версия", IHL, TOS, TTL, "протокол", "адрес источника", "адрес места назначения" в IP-заголовке не передаются в виде части инкапсулированного согласно типу 1 потока IP-пакета 500. Кроме того, первые четыре байта, непосредственно следующие на IP-заголовком, не передаются в виде части инкапсулированного согласно типу 1 потока IP-пакета 500. Эти четыре первых байта соответствуют порту источника и порту места назначения для дейтаграмм протоколов TCP и UDP в качестве примера. Поля порта источника и порта места назначения идентифицируют прикладные программы, выполняемые на станциях. Кроме того, IP-пакет 500, инкапсулированный в соответствии с типом 1 потока, содержит поле данных 512. За полем данных 512 следует поле заполнителя 514 и поле концевика 516 AAL-5 CPCS-PDU. Поле заполнителя 514 может иметь длину от 0 до 47 байтов, а поле концевика 516 равно 8 байтам (четыре 32-битовых слова).

Передаваемый блок IP-пакета, использующего инкапсуляцию согласно типу 1 потока, содержит 1484 байта. Пакеты, принадлежащие к потоку, переадресуемому с использованием инкапсуляции согласно типу 1 потока, передаются в VPI/VCI, определенный в поле метки соответствующего элемента сообщения переадресации типа 1 потока, инкапсулированного в декомпонованном IP-пакете (поле "метка" может быть конфигурировано в соответствии с полями портов источника и места назначения в сообщениях протоколов TCP и UDP).

Тип 2 потока используется для пакетов, переносящих данные между станциями, без учета прикладных задач, выполняемых на станциях. На фиг.10е представлен IP-пакет, инкапсулированный согласно типу 2 потока. IP-пакеты, использующие инкапсуляцию согласно типу 2 потока, декомпонуются, и

выбранные части декомпонованного IP-пакета инкапсулируются непосредственно в полезную нагрузку AAL-5 CPCS-PDU, без приписывания заголовка LLC/SNAP. Инкапсулированный согласно типу 2 потока IP-пакет 520 содержит 16-битовое поле 522 полной длины и 16-битовое поле 524 идентификации из IP-заголовка декомпонованного IP-пакета в качестве первого 32-битового слова. Значение поля 522 полной длины не изменяется, а сохраняет полную длину IP-пакета до декомпоновки. Инкапсулированный согласно типу 2 потока IP-пакет 520 также содержит 8-битовое поле 526 флага, 12-битовое поле 528 смещения фрагмента и 16-битовое поле 530 контрольной суммы из IP-заголовка декомпонованного IP-пакета в качестве второго 32-битового слова. Передаваемое значение поля 530 контрольной суммы представляет собой значение контрольной суммы, которое было бы вычислено для полного IP-заголовка, если бы поле TTL было установлено в нуль. Поля "версия", IHL, TOS, TTL, "протокол", "адрес источника", "адрес места назначения" в IP-заголовке не передаются в виде части инкапсулированного согласно типу 2 потока IP-пакета 520. В отличие от инкапсуляции согласно типу 1 потока, первые четыре байта, непосредственно следующие на IP-заголовком, передаются в виде части инкапсулированного согласно типу 2 потока IP-пакета 520. Кроме того, IP-пакет 520, инкапсулированный в соответствии с типом 2 потока, содержит поле данных 532. За полем данных 532 следует поле заполнителя 534 и поле концевика 536 AAL-5 CPCS-PDU. Поле заполнителя 534 может иметь длину от 0 до 47 байтов, а поле концевика 536 равно 8 байтам (четыре 32-битовых слова). Передаваемый блок IP-пакета, использующего инкапсуляцию согласно типу 2 потока, содержит 1484 байта. Пакеты, принадлежащие к потоку, переадресуемому с использованием инкапсуляции согласно типу 2 потока, передаются в VPI/VCI, определенный в поле метки соответствующего элемента сообщения переадресации типа 2 потока, инкапсулированного в декомпонованном IP-пакете.

Для инкапсуляции согласно типу 0 потока, типу 1 потока и типу 2 потока системный узел, принимающий сообщение переадресации, переданное узлом нисходящей линии, запоминает удаленные поля и связывает эти поля с виртуальным каналом ATM путем ATM метки, обеспечивая кэшированную информацию доступа для переадресованных пакетов, как описано выше.

В. Протокол GSMP

Системное программное обеспечение также использует протокол GSMP для установления связи по ATM линии передачи данных между контроллером коммутации и аппаратными средствами ATM коммутации базового коммутационного блока системы, тем самым обеспечивая коммутацию на уровне 2, если это возможно, и маршрутизацию согласно протоколу IP на уровне 3, если это необходимо. В частности, протокол GSMP представляет собой универсальный асимметричный протокол для

управления ATM коммутатором и реализуется в виртуальном канале, установленном при инициализации в линии передачи ATM данных между контроллером коммутаций и ATM коммутатором. Один контроллер коммутации может использовать множество конкретизаций протокола GSMP по отдельным виртуальным каналам для управления множеством ATM коммутаторов. Протокол GSMP также включает в себя протокол близости GSMP, который используется для синхронизации состояний в линии передачи ATM данных между контроллером коммутации и ATM коммутатором, для определения идентификации узла на другом конце линии связи и для определения изменения в идентификации этого узла.

Протокол GSMP позволяет контроллеру коммутации устанавливать и отменять соединения через ATM коммутатор, добавлять и исключать элементы в соединении от одной точки к множеству точек, управлять портами коммутатора, запрашивать информацию конфигурирования и статистические данные. Протокол GSMP также позволяет ATM коммутатору информировать контроллер коммутации о таких событиях, как отказ в линии связи.

Как упомянуто выше, протокол GSMP представляет собой протокол типа "ведущий-подчиненный". Контроллер коммутации выдает сообщения запроса на коммутатор. Каждое сообщение запроса показывает, требуется ли ответ от коммутатора, и содержит идентификатор операции для обеспечения того, чтобы ответ был связан с конкретным запросом. Коммутатор отвечает ответными сообщениями, указывающими успех или неуспех операции. В рассматриваемом варианте осуществления протокол GSMP предусматривает пять классов сообщений: "управление соединением", "управление портом", "Статистика", "конфигурирование", "событие". За исключением класса сообщений "событие", остальные четыре класса представляют собой классы сообщений типа "запрос-ответ", причем каждый имеет формат для сообщения запроса и формат для ответа при успешной операции. Если только не указано иное, сообщение ответа при неуспешной операции является тем же самым, что и сообщение запроса, которое обусловило этот неуспех, за исключением того, что поле "код" указывает характер такого неуспеха операции. Помимо четырех классов сообщений типа "запрос-ответ", протокол GSMP включает класс сообщений "событие", который позволяет коммутатору генерировать асинхронные сообщения "событие" для информирования контроллера коммутации об асинхронных событиях. Поскольку сообщения "событие" не подтверждаются при приеме контроллером коммутации, сообщения "событие" имеют один формат. В рассматриваемом варианте осуществления имеется множество различных типов сообщений, т.е. функций сообщений протокола GSMP. Каждый из пяти классов сообщений протокола GSMP, за исключением "управления портом", имеет ряд различных типов сообщений.

Протокол GSMP включает сообщение протокола близости GSMP, которому

назначается конкретный тип сообщения. Протокол близости GSMP используется для установления синхронизации в линии передачи ATM данных и поддерживает квитирование. За исключением сообщений протокола близости GSMP никакие другие сообщения GSMP не могут передаваться по линии передачи ATM данных до тех пор, пока протокол близости GSMP не установит синхронизации состояний. Все сообщения GSMP, принятые по линии передачи ATM данных, которые в текущий момент не достигли синхронизации состояний, отбрасываются.

В настоящем изобретении пакеты протокола GSMP имеют переменную длину и инкапсулированы непосредственно в AAL-5 CPCS-PDU с приписанным заголовком LLC/SNAP, аналогично инкапсуляции по умолчанию IP-пакетов в линиях передачи ATM данных, описанных выше со ссылками на фиг.10b. На фиг.11a показан инкапсулированный пакет 540 протокола GSMP. В основном инкапсуляция по умолчанию предусматривает приписывание заголовка LLC/SNAP к пакету протокола GSMP, который инкапсулирован в составе полезной нагрузки AAL-5 CPCS-PDU. Инкапсулированный по умолчанию GSMP-пакет 540 содержит заголовок LLC/SNAP (24-битовое поле 542 LLC, за которым следует 8-битовая часть заголовка 544 SNAP в первом 32-битовом слове, и оставшаяся 32-битовая часть заголовка 544 SNAP), GSMP сообщение 546 (которое имеет длину целого кратного 32-битового слова), поле 548 заполнителя и конечное поле 550 AAL-5 CPCS-PDU. Поле заполнителя 548 может иметь длину от 0 до 47 байтов, поле конечника 550 равно 8 байтам (четыре 32-битовых слова). Передаваемый блок GSMP сообщения 546, использующего инкапсуляцию по умолчанию, равен 1500 байтов. Пакеты, использующие инкапсуляцию по умолчанию, передаются к установленному по умолчанию VPI/VCI, т.е. к установленному по умолчанию виртуальному каналу.

На фиг.11b показана структура сообщения 522 протокола близости GSMP, которое может содержаться в поле 546 сообщения протокола GSMP инкапсулированного пакета 540 протокола GSMP по фиг.11a. Как показано на фиг.11b, сообщение 522 протокола близости GSMP содержит (в порядке от старшего бита к младшему биту) следующие поля: 8-битовое поле "версия" 554, 8-битовое поле "тип сообщения" 556, 8-битовое поле "результат" 558 и 8-битовое поле "код" 560 в качестве первого 32-битового слова; поле 562 "экземпляр передатчика" в качестве второго 32-битового слова; поле 564 "порт передатчика" в качестве третьего 32-битового слова; поле 566 "имя передатчика" в виде следующих 48 битов; поле 568 "имя приемника" в виде следующих 48 битов; поле 570 "порт приемника" в виде следующих 32 битов; и поле 572 "экземпляр приемника" в виде следующих 32 битов. При обсуждении сообщений протокола GSMP под понятием "передатчик" понимается узел, который передает сообщение протокола GSMP по линии передачи ATM данных. Таким узлом может быть контроллер коммутации или ATM коммутатор.

В сообщении 552 протокола близости GSMP поле 554 "версия" определяет версию протокола GSMP, которая используется в данный момент времени. Поле 554 "тип сообщения" устанавливается на конкретное значение (Тип сообщения = 96) для определения сообщения протокола GSMP в качестве сообщения протокола близости GSMP. Поле 556 "результат", не используемое для сообщений протокола близости GSMP, устанавливается в нуль передающим узлом и игнорируется узлом, принимающим сообщение протокола близости GSMP.

Поле 560 кода для сообщений протокола близости GSMP определяет функцию сообщения. В рассматриваемом варианте осуществления имеются четыре возможных значения поля 560 "Code", т.е. функции сообщений протокола близости GSMP: SYN (сообщение синхронизации, Code= 0), SYNACK (сообщение подтверждения синхронизации, Code=1), RSTACK (сообщение подтверждения сброса, Code=2), ACK (сообщение подтверждения, Code=3). В каждом системном узле необходим таймер для периодического генерирования сообщений SYN, SYNACK, ACK протокола GSMP. Для целей протокола близости GSMP конкретный узел имеет три возможных состояния для конкретной линии связи: SYNSENT (сообщение синхронизации передано), SYNRCVD (сообщение синхронизации принято), ESTAB (синхронизация установлена). Синхронизация состояний в линии связи (когда узел достигает состояния ESTAB для данной линии) необходима, прежде чем узлы смогут передавать сообщения протокола GSMP, которые не являются сообщениями протокола близости GSMP. В рассматриваемом варианте осуществления период таймера равен 1 секунде, но могут быть определены и другие длительности периодов таймера. Если значение таймера истекло, и системный узел находится в состоянии SYNSENT, то системный узел сбрасывает таймер и передает сообщение SYN протокола близости GSMP. Если значение таймера истекло, и системный узел находится в состоянии SYNRCVD, то системный узел сбрасывает таймер и передает сообщение SYNACK протокола близости GSMP. Если значение таймера истекло и системный узел находится в состоянии ESTAB, то системный узел сбрасывает таймер и передает сообщение ACK протокола близости GSMP.

В сообщениях SYN, SYNACK, ACK протокола близости GSMP экземпляр передатчика 562 представляет собой "номер экземпляра" передающего узла линии связи. Указывая конкретный экземпляр линии, номер экземпляра представляет 32-битовое ненулевое число, для которого гарантирована его уникальность в пределах ближайшего прошлого и его изменение, если линия или системный узел вводятся в действие после отказа или если идентификация узла на другом конце линии связи изменяется. Соответственно каждая линия имеет свой собственный уникальный номер экземпляра. Экземпляр передатчика 562 используется для определения того, когда линия вновь стала работоспособной после отказа, или когда идентификация узла на другом конце линии

передачи ATM данных изменилась. Для сообщения RSTACK протокола близости GSMP экземпляр передатчика 562 устанавливается на значение поля 572 экземпляра приемника из входящего сообщения протокола близости GSMP, которое обусловило формирование сообщения RSTACK.

В сообщениях SYN, SYNACK, ACK протокола близости GSMP поле 564 порта передатчика представляет собой локальный номер порта линии связи, по которой передается сообщение протокола GSMP. Как указано выше, номера портов представляют собой локально присваиваемые 32-битовые значения. В сообщении RSTACK протокола близости GSMP поле 564 порта передатчика устанавливается на значение поля 570 приемника из входящего сообщения протокола близости GSMP, которое вызвало генерирование сообщения RSTACK.

Для сообщений SYN, SYNACK, ACK протокола близости GSMP поле 566 имени передатчика представляет собой имя передающего узла. 48-битовое поле 566 имени передатчика является уникальным в операционном контексте базового коммутационного блока. Например, для поля "имя передатчика" может быть использован адрес IEEE 802 MAC. Для сообщения RSTACK протокола близости GSMP поле 566 имени передатчика устанавливается на значение поля 566 имени приемника из входящего сообщения протокола близости GSMP, которое вызвало генерирование сообщения RSTACK.

Для сообщений SYN, SYNACK, ACK протокола близости GSMP поле 568 имени приемника представляет собой имя узла, который, как представляется передающему узлу, находится на другом конце линии передачи ATM данных. Если в передающем узле не известно имя этого узла, то поле 568 имени приемника устанавливается на нуль. Для сообщения RSTACK протокола близости GSMP поле 568 имени приемника устанавливается на значение поля 566 имени передатчика из входящего сообщения протокола близости GSMP, которое вызвало генерирование сообщения RSTACK.

Для сообщений SYN, SYNACK, ACK протокола близости GSMP поле 570 порта приемника представляет собой то, что, как представляется передающему узлу, является локальным номером порта, выделенным для линии узлом на другом конце линии передачи ATM данных. Если в передающем узле не известен номер порта этого узла, то поле 570 порта приемника устанавливается на нуль. Для сообщения RSTACK протокола близости GSMP поле 570 порта приемника устанавливается на значение поля 564 порта передатчика из входящего сообщения протокола близости GSMP, которое вызвало генерирование сообщения RSTACK.

Для сообщений SYN, SYNACK, ACK протокола близости GSMP поле 572 экземпляра приемника представляет собой то, что, как представляется передающему узлу, является текущим номером экземпляра, выделенным для линии узлом на другом конце линии передачи. Если в передающем узле не известен текущий номер экземпляра на другом конце линии, то поле 572 экземпляра приемника устанавливается на

нуль. Для сообщения RSTACK протокола близости GSMP поле 572 экземпляра приемника устанавливается на значение поля 562 экземпляра передатчика из входящего сообщения протокола близости GSMP, которое вызвало генерирование сообщения RSTACK.

На фиг.11с представлена диаграмма, иллюстрирующая работу передающего узла после приема входящего сообщения протокола близости GSMP. После запуска системы передающий узел принимает пакет протокола близости GSMP (этап 582). На этапе 584 передающий узел определяет, является ли входящее сообщение протокола близости GSMP сообщением RSTACK. Если входящее сообщение протокола близости GSMP не является сообщением RSTACK (например, сообщение SYN, SYNACK, ACK), то передающий узел работает в соответствии с тем, как показано на диаграмме состояний на фиг.11d. Если входящее сообщение протокола близости GSMP является сообщением RSTACK, то передающий узел проверяет на этапе 584, совпадают ли значения полей экземпляра передатчика, порта передатчика и имени передатчика во входящем сообщении RSTACK со значениями, запомненными из предыдущего сообщения посредством операции обновления верификатора равного по положению. Для протокола близости GSMP операция обновления верификатора равного по положению определяется как запоминание значений полей экземпляра передатчика, порта передатчика и имени передатчика из сообщения SYN или SYNACK, принятого от узла на другом конце линии. Если на этапе 584 установлено, что указанные значения совпадают, то передающий узел на этапе 586 определяет, совпадают ли значения полей экземпляра приемника, порта приемника и имени приемника во входящем сообщении RSTACK со значениями экземпляра приемника, порта приемника и имени приемника, используемыми для всех сообщений SYN, SYNACK или ACK, передаваемых с порта, на который было принято входящее сообщение RSTACK. Если на этапе 586 установлено, что указанные значения совпадают, то передающий узел на этапе 588 определяет, находится ли передающий узел в состоянии SYNSENT. Если передающий узел не находится в состоянии SYNSENT, то передающий узел осуществляет установку линии в исходное состояние на этапе 590. Если на этапе 584 или на этапе 586 установлено, что значения не совпадают, или передающий узел не находится в состоянии SYNSENT, то передающий узел на этапе 592 отбрасывает входящее сообщение RSTACK и ожидает прихода другого пакета. Соответственно, когда в передающий узел приходит сообщение RSTACK протокола близости GSMP, передающий узел устанавливает линию в исходное состояние, как показано этапами 594, 596, 598, 600. На этапе 594 передающий узел генерирует новый номер экземпляра для линии. Затем передающий узел на этапе 596 отменяет (т.е. устанавливает в нуль) запомненные значения полей экземпляра передатчика, порта передатчика и имени передатчика, которые были запомнены в процессе операции

обновления верификатора равного по положению. На этапе 598 передающий узел передает сообщение SYN протокола близости GSMP и на этапе 600 входит в состояние SYNSENT. Передающий узел затем принимает другой пакет для его обработки.

На фиг.11d представлена диаграмма состояний, иллюстрирующая работу передающего узла, когда входящее сообщение протокола близости GSMP не является сообщением RSTACK. Для последующего описания фиг.11d условие "%B" определяется следующим образом: поля экземпляра передатчика, порта передатчика и имени передатчика во входящем сообщении совпадают со значениями, запомненными из предыдущего сообщения посредством операции обновления верификатора равного по положению. Условие "%C" на фиг.11d определяется следующим образом: экземпляр приемника, порт приемника и имя приемника во входящем сообщении совпадают со значениями экземпляра передатчика, порта передатчика и имени передатчика, используемыми в настоящее время для исходящих сообщений SYN, SYNACK, ACK. На фиг. 11d условие "A" означает, что передающий узел принимает входящее сообщение SYNACK протокола близости GSMP и что условие "%C" удовлетворено. Условие "B" означает, что передающий узел принимает входящее сообщение SYNACK протокола близости GSMP и что условие "%C" не удовлетворено. Условие "C" означает, что передающий узел принимает входящее сообщение ACK протокола близости GSMP, и что условия "%B" и "%C" удовлетворены. И условие "D" означает, что передающий узел принимает входящее сообщение ACK протокола близости GSMP, и что условия "%B" и "%C" не удовлетворены.

Если передающий узел находится в состоянии 602 SYNSENT и принимает входящее сообщение SYN протокола близости GSMP от равного по положению на другом конце линии связи, то передающий узел выполняет операцию обновления верификатора равного по положению и передает сообщение SYNACK протокола близости GSMP к равному по положению (показано как этап 604). Затем передатчик переходит из состояния 602 SYNSENT в состояние 606 SYNRCVD. Если передатчик принимает входящее сообщение SYN протокола близости GSMP, находясь в состоянии 606 SYNRCVD, то на этапе 604 передатчик выполняет операцию обновления верификатора равного по положению и передает сообщение SYNACK протокола близости GSMP к равному по положению, но остается в состоянии 606 SYNRCVD. Если передатчик находится в состоянии 606 SYNRCVD и либо состояние B, либо состояние D удовлетворено, то передатчик передает сообщение RSTACK протокола близости GSMP к равному по положению (показано как этап 608) и остается в состоянии 606 SYNRCVD. Если передатчик находится в состоянии 606 SYNRCVD и состояние C удовлетворено, то передатчик передает сообщение ACK протокола близости GSMP к равному по положению (показано как этап 610) и переходит в состояние 612

ESTAB. Если передатчик находится в состоянии 606 SYNRCVD и состояние A удовлетворено, то передатчик выполняет операцию идентификатора обновления верификатора равного по положению, передает сообщение ACK протокола близости GSMP к равному по положению (показано как этап 614) и переходит в состояние 612 ESTAB. Передатчик продолжает находиться в состоянии 612 ESTAB, если передатчик принимает сообщение SYN или SYNACK протокола близости GSMP, или если удовлетворено условие C. Если условие D удовлетворено, когда передатчик находится в состоянии 612 ESTAB, то передатчик остается в состоянии 612 ESTAB и передает сообщение RSTACK протокола близости GSMP (показано как этап 608). В состоянии 602 SYNSENT, если либо передатчик принимает сообщение ACK протокола близости GSMP, либо удовлетворено условие B, то передатчик остается в состоянии 602 SYNSENT и передает сообщение RSTACK протокола близости GSMP (этап 608). Если условие A удовлетворено, когда передатчик находится в состоянии 602 SYNSENT, то передатчик выполняет операцию обновления верификатора равного по положению и передает сообщение ACK протокола близости GSMP (этап 614) и переходит в состояние 612 ESTAB.

Помимо сообщений протокола близости GSMP другие типы сообщений 546 протокола GSMP включают сообщения управления соединением (CM), которые представляют собой сообщения типа "запрос-ответ". В базовом коммутационном блоке контроллер коммутации использует сообщения управления соединением протокола GSMP для установления, отмены, изменения и проверки соединений виртуального канала через ATM коммутатор. Сообщения управления соединением протокола GSMP могут выдаваться независимо от статуса порта коммутации, и соединения могут устанавливаться или отменяться, когда порт коммутатора задействован, неработоспособен или иным образом недоступен. Сообщения управления соединением включают следующие: "добавить переход", "удалить переход", "удалить дерево", "проверить дерево", "удалить все", "переместить корень", "переместить переход". Как отмечено выше, соединение виртуального канала является односторонним и включает в себя входной виртуальный канал и, по меньшей мере, один выходной виртуальный канал или переход ("ветвь"). Это означает, что одностороннее виртуальное соединение имеет одну выходную ветвь, а многостороннее виртуальное соединение имеет две или более выходных ветви.

Сообщение "добавить переход" представляет собой сообщение управления соединением протокола GSMP, используемое для установления соединения виртуального канала или для добавления дополнительной ветви к имеющемуся соединению виртуального канала. В рассматриваемом варианте осуществления изобретения не делается различий между односторонними и многосторонними соединениями. Первое сообщение "добавить переход" для конкретного входного порта, входного

идентификатора виртуального маршрута и входного идентификатора виртуального канала устанавливает однократное соединение. Второе сообщение "добавить переход" с теми же самыми значениями входного порта, входного идентификатора виртуального маршрута и входного идентификатора виртуального канала преобразует однократное соединение в многократное соединение путем добавления другой выходной ветви. Последующие выходные ветви могут добавляться аналогичным образом с помощью последующих сообщений "добавить переход". Кроме того, сообщение "добавить переход" может использоваться для проверки состояния соединения, запомненного в ATM коммутаторе. Сообщение "удалить переход" представляет собой сообщение управления соединением протокола GSMP, используемое для удаления соединения виртуального канала. Например, использование сообщения "удалить переход" для многократного соединения виртуального канала с двумя ветвями удаляет одну ветвь, преобразуя многократное соединение в однократное соединение. Сообщение "удалить переход" может также использоваться для удаления соединения путем удаления последней ветви в соединении виртуального канала. Еще одно сообщение протокола GSMP сообщение "удалить дерево", используется для удаления целого виртуального соединения путем удаления всех оставшихся ветвей соединения. Сообщение "проверить дерево" представляет собой сообщение протокола GSMP, используемое для проверки числа ветвей в соединении виртуального канала. Сообщение "удалить все" представляет собой сообщение протокола GSMP, используемое для удаления всех соединений для входного порта коммутатора. Сообщение "переместить корень" представляет собой сообщение протокола GSMP, используемое для перемещения всего дерева виртуальных соединений от текущего входного порта, входного идентификатора виртуального маршрута и входного идентификатора виртуального канала к новым значениям входного порта, входного идентификатора виртуального маршрута и входного идентификатора виртуального канала. Еще одно сообщение протокола GSMP, сообщение "переместить переход" используется для перемещения одной выходной ветви соединения виртуального канала от его текущих значений выходного порта, выходного идентификатора виртуального маршрута и выходного идентификатора виртуального канала к новым значениям выходного порта, выходного идентификатора виртуального маршрута и выходного идентификатора виртуального канала для того же самого соединения виртуального канала.

На фиг. 12 показана структура обобщенного сообщения 620 управления соединением протокола GSMP, используемого в качестве запроса и ответа для сообщений "добавить переход", "удалить переход", "удалить дерево", "проверить дерево", "удалить все". Сообщение 620 управления соединением протокола GSMP может содержаться в поле 546 сообщения протокола GSMP инкапсулированного пакета

540 протокола GSMP, показанного на фиг.11a. Как показано на фиг.12, обобщенное сообщение 520 управления соединением протокола GSMP содержит (в порядке от старшего бита к младшему биту) следующие поля: 8-битовое поле 622 "версия", 8-битовое поле 624 "тип сообщения", 8-битовое поле 626 "результат", 8-битовое поле 628 "код", 32-битовое поле 630 "идентификатор операции", 32-битовое поле 632 "номер сеанса порта", 32-битовое поле 634 "входной порт", 32-битовое поле "входная метка", которое включает 4-битовое слово 636, установленное на нуль; 12-битовое поле 638 "входной VPI", 16-битовое поле 640 "входной VCI", 32-битовое поле 642 "выходной порт", 32-битовое поле "выходная метка", которое включает 8-битовое слово 644, установленное на нуль; 12-битовое поле 646 "выходной VPI", 16-битовое поле 648 "выходной VCI", 16-битовое поле 650 "число переходов", 8-битовое резервное поле 652 и 8-битовое поле 654 "приоритет".

За исключением сообщений протокола близости GSMP все сообщения протокола GSMP включают поле 622 "версия", поле 624 "тип сообщения", поле 626 "результат", поле 628 "код", поле 630 "идентификатор операции", которые используются принципиально одним и тем же способом. Например, поле 622 2Версия" в сообщении протокола GSMP определяет версию протокола GSMP, которая используется в настоящее время. Поле 624 "тип сообщения" устанавливается на конкретное значение для определения типа сообщения протокола GSMP. Например, сообщению "добавить переход" управления соединением протокола GSMP присваивается конкретное значение для поля 624 "тип сообщения", а другим типам сообщений присваиваются другие конкретные значения.

Для сообщения протокола GSMP, которое является сообщением запроса, поле 626 "результат" указывает, требуется ли ответ на сообщение запроса, если результат выполнения является успешным. Поле 626 "результат" в сообщении запроса может содержать значения для NoSuccessAck (указывая, что при успешном результате не требуется ответа) или значения для AckAll (указывая, что при успешном результате требуется ответ). Для некоторых типов сообщений запроса протокола GSMP AckAll устанавливается по умолчанию, а значение NoSuccessAck в поле "результат" 626 игнорируется. Для сообщения протокола GSMP, которое является сообщением ответа, поле 626 "результат" может содержать значения для успешного результата (указывая, что запрос был успешным), или отсутствия успешного результата (указывая, что результат был неудачным). Сообщение успешного ответа протокола GSMP не передается до тех пор, пока запрос не будет успешно завершен. Сообщение успешного ответа протокола GSMP является копией соответствующего сообщения запроса, возвращенного с полем 626 "результат", указывающим успешный результат. Для сообщения запроса протокола GSMP, которое не имеет успешного результата, формируется ответное сообщение, указывающее сбой. Ответное сообщение протокола GSMP, указывающее сбой, представляет собой

копию соответствующего запросного сообщения, возвращенного с полем 626 "результат", указывающим сбой. Коммутатор, выдающий ответное сообщение протокола GSMP, указывающее наличие сбоя, в ответ на неудачный результат сообщения запроса, не модифицирует состояние соединения в коммутаторе.

В сообщении протокола GSMP, которое является сообщением запроса, поле 628 "код" может обеспечивать дополнительную информацию относительно результата. Например, поле 628 "код" может содержать код ошибки, определяющий тип ошибки, вызвавшей сбой. Следует иметь в виду, что множество различных типов кодов могут быть определены для использования в поле 626 "код". Примерами кодов ошибок могут служить следующие: отказ, специфический для конкретного типа сообщения; неопределенная причина, не перекрываемая другими типами кодов сбоев; недействительное сообщение запроса; определенное сообщение запроса не выполняется на данном коммутаторе; недействительный номер сеанса порта; по меньшей мере один определенный порт не существует; по меньшей мере один определенный порт неисправен; по меньшей мере один определенный VPI/VCi находится вне диапазона для по меньшей мере одного определенного порта; определенное соединение не существует; определенная выходная ветвь не существует; определенная выходная ветвь уже установлена для конкретного многократного соединения на определенном выходном порте; достигнут максимальный предел для многократных соединений, поддерживаемых коммутатором; достигнут максимальный предел для ветвей, который может поддерживаться определенным многократным соединением; общая проблема, связанная с возможностями многократного соединения, поддерживаемого коммутатором. Разумеется, могут быть использованы и другие коды. Кроме того, поле 628 "код" может обеспечивать другую информацию в ответном сообщении успешного результата или сообщении типа "событие". Поле 628 "код" не используется в сообщениях запроса протокола GSMP и устанавливается в нуль.

Поле 630 "идентификатор операции" используется для того, чтобы увязать сообщение запроса протокола GSMP с сообщением ответа протокола GSMP. В сообщении запроса протокола GSMP контроллер коммутации выбирает значение идентификатора операции для поля 630. В сообщении ответа протокола GSMP значение идентификатора операции для поля 630 устанавливается на значение идентификатора операции из сообщения запроса, на которое отвечает соответствующее сообщение ответа протокола GSMP. Поскольку сообщение типа "событие" протокола GSMP не требует ответа, то в нем поле 630 идентификатора операции установлено в нуль.

Следует иметь в виду, что приведенное выше описание полей "версия", "тип сообщения", "результат", "код", "идентификатор операции" применимо ко всем сообщениям протокола GSMP, за исключением сообщений протокола близости GSMP. Отличия от общего описания

приводятся в той мере, как это необходимо.

Для сообщений управления соединением протокола GSMP поле 632 "номер сеанса порта" обеспечивает номер сеанса для входного порта. В частности, значение поля 632 "номер сеанса порта" указывает номер сеанса порта для входного порта коммутатора, указанного в поле 634 "входной порт". Каждый порт коммутатора поддерживает номер сеанса порта, присвоенный коммутатором. Номер сеанса порта остается неизменным, пока порт находится в работоспособном состоянии. Однако если порт становится работоспособным после отказа, то генерируется новый отличающийся номер сеанса порта. Предпочтительно новый номер сеанса порта генерируют случайным образом. Если контроллер коммутации передает сообщение запроса управления соединением протокола GSMP, которое имеет недействительное поле 632 "номер сеанса порта", то коммутатор режетирует сообщение запроса управления соединением протокола GSMP путем отправки ответного сообщения управления соединением, указывающего наличие сбоя, в котором поле 628 "код" указывает недействительный номер сеанса порта, вызвавший сбой. Текущий номер сеанса порта может быть получен с использованием сообщения конфигурирования протокола GSMP.

В сообщении управления соединением протокола GSMP поле 634 "входной порт" указывает входной порт коммутатора с использованием 32-битового значения, присвоенного коммутатором. Поле 638 "входной VPI" идентифицирует виртуальный ATM маршрут, поступающий на входной порт коммутатора, указанный полем 634 "входной порт", а поле 640 "входной VCI" идентифицирует виртуальный ATM канал, поступающий по виртуальному маршруту, указанному полем 638 "входной VPI".

В сообщении управления соединением протокола GSMP поле 642 "выходной порт" указывает выходной порт коммутатора с использованием 32-битового значения, присвоенного коммутатором. Поле 646 "выходной VPI" идентифицирует виртуальный ATM маршрут, выходящий из выходного порта коммутатора, указанного полем 642 "выходной порт", а поле 648 "выходной VCI" идентифицирует виртуальный ATM канал, выходящий по виртуальному маршруту, указанному полем 646 "выходной VPI".

Для сообщения управления соединением протокола GSMP поле 650 "число переходов" дает число выходных ветвей в соединении виртуального канала. Поле 650 используется в сообщении "проверить дерево" управления соединением протокола GSMP. Для всех сообщений управления соединением протокола GSMP поле 650 устанавливается на нуль передающим узлом и игнорируется приемным узлом. В рассматриваемом варианте осуществления резервное поле 652, которое не используется для сообщений управления соединением протокола GSMP, устанавливается на нуль передающим узлом и игнорируется приемным узлом.

Поле 654 "приоритет" в сообщении управления соединением протокола GSMP определяет приоритет соединения. Наивысший приоритет обозначается нулем, а

самый низкий приоритет обозначается q-1, где q - число приоритетов, которое может поддерживать выходной порт коммутатора. Число q для каждого выходного порта коммутатора может быть получено из сообщения конфигурирования порта GSMP. Каждое соединение виртуального канала может быть установлено с определенным качеством обслуживания, путем присвоения ему приоритета при установлении. Для соединений виртуального канала, которые совместно используют один и тот же выходной порт, ATM элемент данных в соединении с более высоким приоритетом с большей вероятностью будет отправлен из коммутатора, чем ATM элемент данных в соединении с менее высоким приоритетом, если оба они присутствуют в коммутаторе в одно и то же время. Поле 654 "приоритет" используется в сообщениях "добавить переход" и "переместить переход" управления соединением протокола GSMP. Если сообщение запроса управления соединением протокола GSMP ("добавить переход" или "переместить переход") имеет значение в поле 654 "приоритет", которое не поддерживается коммутатором, то коммутатор присваивает наивысший приоритет, поддержку которого он обеспечивает. В других сообщениях управления соединением протокола GSMP поле 654 "приоритет" устанавливается на нуль передающим узлом и игнорируется приемным узлом.

Сообщение "добавить переход" является сообщением управления соединением протокола GSMP, используемым для установления соединения виртуального канала или для добавления дополнительной ветви к существующему соединению виртуального канала. Соединение определяется полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI", а выходная ветвь определяется полем 642 "выходной порт", полем 646 "выходной VPI" и полем 648 "выходной VCI", причем приоритет соединения определяется полем 654 "приоритет". Кроме того, сообщение "добавить переход" может использоваться для проверки состояния соединения, запомненного в ATM коммутаторе. На фиг.13а показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "добавить переход" протокола GSMP от контроллера коммутации. На этапе 600 контроллер коммутации передает сообщение запроса "добавить переход" протокола GSMP, которое принимается ATM коммутатором. ATM коммутатор на этапе 662 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI" принятого сообщения запроса "добавить переход". Если коммутатор на этапе 622 определяет, что такое виртуальное соединение не существует, то на этапе 664 ATM коммутатор устанавливает соединение, как определено в сообщении запроса "добавить переход". Если коммутатор на этапе 622 определяет, что такое виртуальное соединение существует, то на этапе 666 ATM коммутатор определяет, существует ли в коммутаторе выходная ветвь,

как определено полем 642 "выходной порт", полем 646 "выходной VPI" и полем 648 "выходной VCI" принятого сообщения запроса "добавить переход". Если определено, что такой выходной ветви нет, то ATM коммутатор на этапе 668 добавляет новую выходную ветвь, как определено в сообщении запроса "добавить переход". После этапов 664 или 668 коммутатор на этапе 670 определяет, осуществлена ли указанная операция успешно. Если операция была неуспешной, то ATM коммутатор на этапе 672 посылает контроллеру коммутации сообщение ответа "добавить переход", которое является копией сообщения запроса "добавить переход" с полем 626 "результат", указывающим сбой. Сообщение ответа "добавить переход" может также определить тип отказа с помощью соответствующего кода в поле 628 "код". Если на этапе 670 определено, что операция успешно завершена, то ATM коммутатор на этапе 674 проверяет поле 626 "результат" сообщения запроса "добавить переход", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает AckAll, то ATM коммутатор на этапе 676 посылает ответ с индикацией успешной операции контроллеру коммутации. Сообщение ответа "добавить переход" является копией сообщения запроса "добавить переход" с полем 626 "результат", указывающим успешный результат. Если коммутатор на этапе 666 определяет, что выходная ветвь, определенная в сообщении запроса "добавить переход", уже существует, то коммутатор на этапе 680 проверяет, является ли приоритет, определенный в поле "приоритет" 654 сообщения запроса, отличающимся от текущего приоритета данной выходной ветви. Если коммутатор определил, что запрашиваемый приоритет отличается от текущего приоритета, то коммутатор на этапе 682 изменяет приоритет выходной ветви на тот, который определен в сообщении запроса "добавить переход". Если приоритеты одинаковы, то коммутатор не изменяет приоритет (показано этапом 684).

Сообщение "удалить переход" является сообщением управления соединением протокола GSMP, используемым для удаления соединения виртуального канала, или в случае последней ветви, для удаления соединения виртуального канала. Соединение определяется полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI", а выходная ветвь определяется полем 642 "выходной порт", полем 646 "выходной VPI" и полем 648 "выходной VCI". На фиг.13b показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "удалить переход" протокола GSMP от контроллера коммутации. На этапе 690 контроллер коммутации передает сообщение запроса "удалить переход" протокола GSMP, которое принимается ATM коммутатором. ATM коммутатор на этапе 692 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI" принятого сообщения запроса "удалить переход". Если коммутатор на этапе 622 определяет, что

такое виртуальное соединение существует, то на этапе 694 ATM коммутатор определяет, существует ли в коммутаторе выходная ветвь, как определено полем 642 "выходной порт", полем 646 "выходной VPI" и полем 648 "выходной VCI" принятого сообщения запроса "удалить переход". Если определено, что такая выходная ветвь есть, то ATM коммутатор на этапе 696 удаляет выходную ветвь, как определено в сообщении запроса "удалить переход". После этапа 696 коммутатор на этапе 698 определяет, осуществлена ли указанная операция успешно. Если операция была успешной, то ATM коммутатор на этапе 700 проверяет поле 626 "результат" сообщения запроса "удалить переход", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает AskAll, то ATM коммутатор на этапе 702 посылает ответ с индикацией успешной операции удаления ветви контроллеру коммутации. Сообщение ответа "удалить переход" является копией сообщения запроса "удалить переход" с полем 626 "результат", указывающим успешный результат. Если на этапе 700 определено, что ответ в случае успеха не требуется, то коммутатор не вырабатывает ответного сообщения (этап 704). Если коммутатор на этапе 692 определяет, что соединение, определенное в сообщении запроса "удалить переход", не существует, или коммутатор на этапе 692 определяет, что выходная ветвь, определенная в сообщении запроса "удалить переход", не существует, или коммутатор на этапе 698 определяет, что операция удаления неуспешна, то коммутатор на этапе 706 передает сообщение ответа "удалить переход" с индикацией сбоя контроллеру коммутации с указанием соответствующего кода сбоя. Сообщение ответа "удалить переход" с индикацией сбоя является копией принятого сообщения запроса "удалить переход" с полем 626 "результат", указывающим сбой, и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код".

Сообщение "удалить дерево" используется для удаления всего соединения виртуального канала путем удаления всех оставшихся ветвей соединения. Соединение определяется полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI". Поле 642 "выходной порт", поле 646 "выходной VPI" и поле 648 "выходной VCI" в сообщении "удалить дерево" не используются. На фиг.13с показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "удалить дерево" протокола GSMP от контроллера коммутации. На этапе 710 контроллер коммутации передает сообщение запроса "удалить дерево" протокола GSMP, которое принимается ATM коммутатором. ATM коммутатор на этапе 712 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI" принятого сообщения запроса "удалить дерево". Если коммутатор на этапе 712 определяет, что такое соединение виртуального канала существует, то на этапе 714 ATM коммутатор

удаляет соединение (и тем самым все дерево), как определено в сообщении запроса "удалить дерево". После этапа 714 коммутатор на этапе 716 определяет, осуществлена ли указанная операция успешно. Если операция была успешной, то ATM коммутатор на этапе 718 проверяет поле 626 "результат" сообщения запроса "удалить дерево", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает AskAll, то ATM коммутатор на этапе 720 посылает ответ с индикацией успешной операции удаления дерева контроллеру коммутации. Сообщение ответа "удалить дерево" с индикацией успеха является копией принятого сообщения запроса "удалить дерево" с полем 626 "результат", указывающим успешный результат. Если на этапе 716 определено, что ответ в случае успеха не требуется, то коммутатор не вырабатывает ответного сообщения (этап 722). Если коммутатор на этапе 712 определяет, что соединение, определенное в сообщении запроса "удалить дерево", не существует, или коммутатор на этапе 712 определяет, что операция удаления неуспешна, то коммутатор на этапе 724 передает сообщение ответа "удалить дерево" с индикацией сбоя контроллеру коммутации с указанием соответствующего кода сбоя. Сообщение ответа "удалить дерево" с индикацией сбоя является копией принятого сообщения запроса "удалить дерево" с полем 626 "результат", указывающим сбой, и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код".

Сообщение "проверить дерево" используется для проверки числа ветвей соединения виртуального канала. Соединение определяется полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI". Поле 642 "выходной порт", поле 646 "выходной VPI" и поле 648 "выходной VCI" в сообщении "проверить дерево" не используются, устанавливаются в нуль контроллером коммутации и игнорируются коммутатором. На фиг.13d показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "проверить дерево" протокола GSMP от контроллера коммутации. На этапе 730 контроллер коммутации передает сообщение запроса "проверить дерево" протокола GSMP, которое принимается ATM коммутатором. ATM коммутатор на этапе 732 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI" принятого сообщения запроса "проверить дерево". Если коммутатор на этапе 732 определяет, что такое соединение виртуального канала существует, то на этапе 734 ATM коммутатор проверяет действительное число ветвей для конкретного соединения и сравнивает действительное число с тем, которое указано в поле 650 "число ветвей" принятого сообщения запроса "проверить дерево". Если коммутатор на этапе 736 определяет, что указанные числа совпадают, то указанная операция осуществлена успешно. Если операция была успешной, то ATM коммутатор на этапе 738 проверяет поле 626 "результат"

сообщения запроса "проверить дерево", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает AckAll, то ATM коммутатор на этапе 740 посылает ответ с индикацией успешной операции проверки дерева контроллеру коммутации. Сообщение ответа "проверить дерево" с индикацией успеха является копией принятого сообщения запроса "проверить дерево" с полем 626 "результат", указывающим успешный результат. Если на этапе 738 определено, что ответ в случае успеха не требуется, то коммутатор не вырабатывает ответного сообщения (этап 742). Если коммутатор на этапе 732 определяет, что соединение, определенное в сообщении запроса "проверить дерево", не существует, то коммутатор на этапе 744 передает сообщение ответа "проверить дерево" с индикацией сбоя контроллеру коммутации с указанием соответствующего кода сбоя. Сообщение ответа "проверить дерево" с индикацией сбоя является копией принятого сообщения запроса "проверить дерево" с полем 626 "результат", указывающим сбой, и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код". Если коммутатор на этапе 736 определяет, что операция проверки неуспешна, то коммутатор на этапе 746 устанавливает действительный номер ветвей в поле 650 "число ветвей" сообщения ответа "проверить дерево" с индикацией сбоя и передает его к контроллеру коммутации с полем 628 "код", установленным в нуль.

Сообщение "удалить все" используется для удаления всех соединений на входном порте коммутатора. Входной порт определяется полем 634 "входной порт". В сообщении "удалить все" поле 638 "входной VPI", поле 640 "входной VCI", поле 642 "выходной порт", поле 646 "выходной VPI" и поле 648 "выходной VCI" не используются, устанавливаются в нуль контроллером коммутации и игнорируются коммутатором. На фиг.13е показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "удалить все" протокола GSMP от контроллера коммутации. На этапе 750 контроллер коммутации передает сообщение запроса "удалить все" протокола GSMP, которое принимается ATM коммутатором. ATM коммутатор на этапе 752 определяет, существуют ли какие-либо соединения на входном порте коммутатора, как определено полем 634 "входной порт" принятого сообщения запроса "удалить все". Если коммутатор на этапе 752 определяет, что такие соединения существуют, то на этапе 754 ATM коммутатор удаляет все соединения для входного порта коммутатора, как определено в принятом сообщении запроса "удалить все". Затем коммутатор на этапе 756 определяет, осуществлена ли указанная операция успешно. Если операция была успешной, то ATM коммутатор на этапе 758 проверяет поле 626 "результат" сообщения запроса "удалить все", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает AckAll, то ATM коммутатор на этапе 760 посылает ответ с индикацией успешной операции удаления

контроллеру коммутации. Сообщение ответа "удалить все" с индикацией успеха является копией принятого сообщения запроса "удалить все" с полем 626 "результат", указывающим успешный результат. Если на этапе 758 определено, что ответ в случае успеха не требуется, то коммутатор не вырабатывает ответного сообщения (этап 762). Если коммутатор на этапе 752 определяет, что нет соединений для входного порта, определенного в сообщении запроса "удалить все", то коммутатор на этапе 764 передает сообщение ответа "удалить все" с индикацией сбоя контроллеру коммутации с указанием соответствующего кода сбоя. Сообщение ответа "удалить все" с индикацией сбоя является копией принятого сообщения запроса "удалить все" с полем 626 "результат", указывающим сбой, и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код".

Сообщение "переместить корень" представляет собой сообщение управления соединением протокола GSMP, используемое для перемещения всего дерева виртуальных соединений с текущего входного порта, входного VPI и входного VCI к новому входному порту, входному VPI и входному VCI. На фиг.13f показана структура сообщения 770 "переместить корень" управления соединением протокола GSMP, используемого в качестве запроса и ответа. Сообщение 770 "переместить корень" управления соединением протокола GSMP содержит (в порядке от старшего бита к младшему биту) следующие поля: 8-битовое поле 622 "версия", 8-битовое поле 624 "тип сообщения", 8-битовое поле 626 "результат", 8-битовое поле 628 "код", 32-битовое поле 630 "идентификатор операции", 32-битовое поле 632 "номер сеанса порта", 32-битовое поле 772 "старый входной порт"; 4-битовое слово 774, установленное на нуль; 12-битовое поле 776 "старый входной VPI", 16-битовое поле 778 "старый входной VCI", 32-битовое поле 780 "новый входной порт"; 8-битовое слово 782, установленное на нуль; 12-битовое поле 784 "новый входной VPI", 16-битовое поле 786 "новый входной VCI", 32-битовое резервное поле 788. Поле 622 "версия", поле 624 "тип сообщения", поле 626 "результат", поле 628 "код", поле 630 "идентификатор операции", поле 632 "номер сеанса порта" используются принципиально тем же путем, как и для других сообщений управления соединением протокола GSMP, как рассмотрено выше. Резервное поле 788 не используется, устанавливается на нуль передатчиком и игнорируется приемником. В сообщении "переместить корень" текущее соединение виртуального канала определяется полем 772 "старый входной порт"; полем 776 "старый входной VPI", полем 778 "старый входной VCI", а новое соединение виртуального канала определяется полем 780 "новый входной порт"; полем 784 "новый входной VPI" и полем 786 "новый входной VCI". На фиг.13g показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "переместить корень" протокола GSMP от контроллера коммутации. На этапе 790 контроллер коммутации передает сообщение запроса "переместить корень" протокола

GSMP, которое принимается ATM коммутатором. ATM коммутатор на этапе 792 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 772 "старый входной порт", полем 776 "старый входной VPI" и полем 778 "старый входной VCI" принятого сообщения запроса "переместить корень". Если коммутатор на этапе 792 определяет, что такое соединение виртуального канала существует, то на этапе 794 ATM коммутатор является ли соединение виртуального канала, определяемое полем 780 "новый входной порт", полем 784 "новый входной VPI" и полем 786 "новый входной VCI", не назначенным. Если на этапе 794 определено, что такое соединение виртуального канала является не назначенным, то коммутатор на этапе 796 перемещает каждую выходную ветвь существующего соединения виртуального канала для установления нового соединения виртуального канала, как определено в сообщении запроса "переместить корень". После этапа 796 коммутатор на этапе 798 определяет, осуществлена ли указанная операция успешно. Если операция была успешной, то ATM коммутатор на этапе 800 проверяет поле 626 "результат" сообщения запроса "переместить корень", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле

"результат" сообщения запроса указывает AckAll, то ATM коммутатор на этапе 802 посылает ответ с индикацией успешной операции перемещения корня контроллеру коммутации. Сообщение ответа "переместить корень" с индикацией успеха является копией принятого сообщения запроса "переместить корень" с полем 626 "результат", указывающим успешный результат. Если на этапе 800 определено, что ответ в случае успеха не требуется, то коммутатор не вырабатывает ответного сообщения (этап 804). Если коммутатор на этапе 792 определяет, что старое соединение, определенное в сообщении запроса "переместить корень", не существует, то коммутатор на этапе 806 передает сообщение ответа "переместить корень" с индикацией сбоя контроллеру коммутации с указанием соответствующего кода сбоя. Сообщение ответа "переместить корень" с индикацией сбоя является копией принятого сообщения запроса "переместить корень" с полем 626 "результат", указывающим сбой и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код". Если коммутатор на этапе 794 определяет, что новое соединение виртуального канала, определенное в сообщении запроса "переместить корень", является назначенным, то коммутатор не производит никаких изменений в существующих соединениях и устанавливает поле 628 "код" в нуль в сообщении ответа "переместить корень" с индикацией сбоя (этап 808) перед передачей его контроллеру коммутации на этапе 806.

Еще одно сообщение управления соединением протокола GSMP сообщение "переместить ветвь" используется для перемещения одной выходной ветви соединения виртуального канала с ее текущего выходного порта, выходного VPI и выходного VCI к новому выходному порту, новому VPI и новому VCI в том же самом

соединении виртуального канала. На фиг. 13h показана структура сообщения 820 "переместить ветвь" управления соединением протокола GSMP, используемого в качестве запроса и ответа. Сообщение 820 "переместить ветвь" управления соединением протокола GSMP содержит (в порядке от старшего бита к младшему биту) следующие поля: 8-битовое поле 622 "версия", 8-битовое поле 624 "тип сообщения", 8-битовое поле 626 "результат", 8-битовое поле 628 "код", 32-битовое поле 630 "идентификатор операции", 32-битовое поле 632 "номер сеанса порта", 32-битовое поле 634 "входной порт"; 4-битовое слово 636, установленное на нуль; 12-битовое поле 638 "входной VPI", 16-битовое поле 640 "входной VCI", 32-битовое поле 822 "старый выходной порт", 8-битовое слово 824, установленное на нуль; 12-битовое поле 826 "старый выходной VPI", 16-битовое поле 828 "старый выходной VCI", 32-битовое поле 830 "новый выходной порт"; 8-битовое слово 832, установленное на нуль; 12-битовое поле 834 "новый выходной VPI", 16-битовое поле 836 "новый выходной VCI", 24-битовое резервное поле 838, поле 940 "приоритет". Поле 622 "версия", поле 624 "тип сообщения", поле 626 "результат", поле 628 "код", поле 630 "идентификатор операции", поле 632 "номер сеанса порта" используются принципиально тем же путем, как и для других сообщений управления соединением протокола GSMP, как рассмотрено выше. Резервное поле 838 не используется, устанавливается на нуль передатчиком и игнорируется приемником. Поле 940 "приоритет" используется аналогичным образом, как описано выше для поля 654 "приоритет" сообщения управления соединением протокола GSMP. В сообщении "переместить ветвь" соединение виртуального канала определяется полем 634 "входной порт", полем 638 "входной VPI", полем 640 "входной VCI". Старая ветвь соединения виртуального канала определяется полем 822 "старый выходной порт", полем 826 "старый выходной VPI", полем 828 "старый выходной VCI". Новая ветвь соединения виртуального канала определяется полем 830 "новый выходной порт", полем 834 "новый выходной VPI" и полем 836 "новый выходной VCI". На фиг. 13i показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "переместить ветвь" протокола GSMP от контроллера коммутации. На этапе 842 контроллер коммутации передает сообщение запроса "переместить ветвь" протокола GSMP, которое принимается ATM коммутатором. ATM коммутатор на этапе 844 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 634 "входной порт", полем 638 "входной VPI", полем 640 "входной VCI" принятого сообщения запроса "переместить ветвь". Если коммутатор на этапе 844 определяет, что такое соединение виртуального канала существует, то коммутатор на этапе 846 определяет, существует ли в этом соединении виртуального канала старая выходная ветвь, как определено полем 822 "старый выходной порт", полем 826 "старый выходной VPI", полем 828 "старый выходной VCI" принятого

сообщения запроса "переместить ветвь". Если на этапе 846 определено, что такая старая выходная ветвь существует, то коммутатор на этапе 848 добавляет новую выходную ветвь, как определено полем 830 "новый выходной порт", полем 834 "новый выходной VPI" и полем 836 "новый выходной VCI" в сообщении запроса "переместить ветвь" и удаляет старую выходную ветвь, как определено в сообщении запроса "переместить ветвь". После этапа 848 коммутатор на этапе 850 определяет, осуществлена ли указанная операция успешно. Если операция была успешной, то АТМ коммутатор на этапе 852 проверяет поле 626 "результат" сообщения запроса "переместить ветвь", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает AckAll, то АТМ коммутатор на этапе 854 посылает ответ с индикацией успешной операции перемещения ветви контроллеру коммутации. Сообщение ответа "переместить ветвь" с индикацией успеха является копией принятого сообщения запроса "переместить ветвь" с полем 626 "результат", указывающим успешный результат. Если на этапе 852 определено, что ответ в случае успеха не требуется, то коммутатор не вырабатывает ответного сообщения (этап 856). Если коммутатор на этапе 844 определяет, что соединение виртуального канала, определенное в сообщении запроса "переместить ветвь", не существует, или если коммутатор на этапе 846 определяет, что старая ветвь, определенная в сообщении запроса "переместить ветвь", не существует, или если коммутатор на этапе 850 определит, что операция перемещения ветви неуспешна, то коммутатор на этапе 858 не модифицирует никаких состояний соединений и передает на этапе 860 сообщение ответа "переместить ветвь" с индикацией сбоя контроллеру коммутации с указанием соответствующего кода сбоя. Сообщение ответа "переместить ветвь" с индикацией сбоя является копией принятого сообщения запроса "переместить ветвь" с полем 626 "результат", указывающим сбой, и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код".

Обеспечивая управление коммутационными портами, сообщение управления портами протокола GSMP обеспечивает возможность активизации порта для осуществления обслуживания, исключения порта из процедуры обслуживания, закольцовывания или установки в исходное состояние. На фиг.14 представлена структура сообщения 870 управления портом протокола GSMP, используемого в качестве сообщений запроса и ответа. Сообщение 870 управления портом протокола GSMP может содержаться в поле 546 сообщения протокола GSMP инкапсулированного пакета 540 протокола GSMP, как показано на фиг.11а. Согласно фиг.14, сообщение 870 управления портом протокола GSMP содержит (в порядке от старшего бита к младшему) следующие поля: 8-битовое поле 622 "версия", 8-битовое поле 624 "тип сообщения", 8-битовое поле 626 "результат", 8-битовое поле 628 "код", 32-битовое поле 630 "идентификатор операции", 32-битовое поле 872 "порт",

32-битовое поле 874 "номер сеанса порта", 32-битовое поле 876 "номер последовательности события", 8-битовое поле 878 "флаг события", 8-битовое поле 880 "длительность", 16-битовое поле "функция". Поле 622 "версия", поле 624 "тип сообщения", поле 626 "результат", поле 628 "код", поле 630 "идентификатор операции", поле 874 "номер сеанса порта" используются принципиально тем же путем, как и для других сообщений управления соединением протокола GSMP, как рассмотрено выше. Поле 872 "порт" указывает номер порта, для которого применяется сообщение управления портом протокола GSMP. Сообщение управления портом протокола GSMP имеет конкретное поле "тип сообщения" и различные возможные функции, которые могут быть определены в поле 882 "функция". Некоторые из функций сообщения управления портом протокола GSMP включают следующее: функция включения, функция отключения, функция внутреннего закольцовывания, функция внешнего закольцовывания, функция двунаправленного закольцовывания, функция сброса входного порта, функция сброса флагов событий. Каждый коммутационный порт поддерживает номер последовательности события и набор флагов событий (по одному флагу события на каждый тип сообщения "событие"). Номер последовательности события устанавливается в нуль, когда порт инициализируется, и получает приращение каждый раз, когда на этом порте обнаруживается асинхронное событие, оповещающее сообщением "событие", независимо от того, передано ли сообщение "событие" или нет. Если коммутационный порт передает сообщение "событие", он устанавливает соответствующий флаг события на данном порте. Тем самым порту не разрешено передавать другое сообщение "событие" того же самого типа, пока соответствующий флаг события не будет сброшен функцией сброса флагов событий сообщения управления портом протокола GSMP. Использование флагов событий обеспечивает простое управление потоком, предотвращающее перегрузку контроллера коммутации сообщениями "событие" со стороны коммутатора. В сообщении запроса управления портом протокола GSMP поле 876 "номер последовательности события" не используется, устанавливается на нуль контроллером коммутации и игнорируется коммутатором. В сообщении ответа управления портом протокола GSMP с индикацией успеха поле 876 "номер последовательности события" указывает текущее значение номера последовательности события коммутационного порта, определенного в принятом сообщении запроса управления портом протокола GSMP. В сообщении запроса управления портом протокола GSMP с полем 882 "функция", определяющим сброс флагов событий, конкретные биты в поле 878 "флаги событий" могут использоваться для сброса соответствующих флагов событий в порте, определенном полем 872 "порт". В сообщении ответа управления портом протокола GSMP с индикацией успеха, в котором поле 882 "функция" определяет сброс флагов событий, биты поля 878 "флаги

событий" устанавливаются на текущие значения, соответствующие флагам событий для определенного порта, после того как флаги событий, определенные в сообщении запроса, будут сброшены. Путем установки поля "флаги событий" на все нули в сообщении управления портом протокола GSMP с помощью функции сброса флагов событий, контроллер коммутации способен получить текущее состояние флагов событий и текущий номер последовательности события для определенного порта без изменения состояния флагов событий. В других сообщениях управления портом протокола GSMP с другим определенным полем "функция" поле 878 "флаги событий" не используется, устанавливается в нуль контроллером коммутации и игнорируется коммутатором. Поле 880 "длительность" используется только в сообщениях управления портом протокола GSMP с полем 882 "функция", определенным как функция внутреннего закольцовывания, функция внешнего закольцовывания или функция двунаправленного закольцовывания. Поле 880 "длительность" обеспечивает длительность (в секундах), в течение которой сохраняется состояние закольцовывания. По истечении установленного интервала порт, который находился в состоянии закольцовывания, автоматически возвращается в рабочее состояние. В сообщении управления портом с различным определенным полем 882 "функция" поле 880 "длительность" не используется. Оно устанавливается на нуль контроллером коммутации и игнорируется коммутатором. В сообщениях управления портом поле 882 "функция" определяет действие, которое должно быть осуществлено (определенное действие осуществляется независимо от текущего статуса порта). Функция включения вводит порт в рабочее состояние, а функция отключения выводит порт из рабочего состояния. Функция внутреннего закольцовывания выполняет внутреннее закольцовывание (ATM элементы данных, приходящие на выходной порт с коммутатора, закольцовываются через входной порт обратно на коммутатор). Функция внешнего закольцовывания выполняет внешнее закольцовывание (ATM элементы данных, приходящие на входной порт из внешней линии связи, закольцовываются назад в линию связи на физическом уровне, без ввода во входной порт). Функция двунаправленного закольцовывания выполняет как внешнее, так и внутреннее закольцовывание. Функция сброса входного порта устанавливает в исходное состояние входной порт (все соединения, приходящие на конкретный входной порт, удаляются, а аппаратные средства входного и выходного портов повторно инициализируются, так что все значения VPI/VC1 для конкретного входного порта в таблице соединений являются незаполненными). Функция сброса флагов событий сбрасывает флаги событий, как описано выше.

Сообщения статистики протокола GSMP обеспечивают возможность контроллеру коммутации запрашивать значения различных счетчиков аппаратных средств, связанных с входными и выходными портами коммутатора и с виртуальными каналами. Предусмотрены

два класса сообщений статистики: сообщения активности виртуального канала и сообщения статистики порта и виртуального канала. Сообщение активности виртуального канала используется для определения того, осуществлялась ли передача графика по одному или нескольким определенным виртуальным каналам. Сообщение активности виртуального канала содержит одну или несколько записей активности виртуального канала. Каждая запись активности виртуального канала используется для запроса и возврата в ответ информации активности, относящейся к одному определенному виртуальному соединению. Если коммутатор поддерживает трафик, учитываемый на каждое виртуальное соединение, то текущее значение счетчика трафика для каждого конкретного виртуального соединения возвращается в поле "счет трафика виртуального канала" записи активности виртуального канала. Текущее значение счета трафика сравнивается с предыдущими значениями для каждого определенного виртуального соединения для определения того, было ли виртуальное соединение активным в соответствующий период. Если коммутатор поддерживает определение трафика, приходящегося на виртуальное соединение некоторым иным образом, отличным от подсчета трафика, то результат может указываться для виртуального соединения с использованием поля "флаг" в записи активности виртуального канала. Сообщения статистики порта и виртуального канала используются для опроса различных счетчиков, определенных для трафика и ошибок для порта и виртуального канала. Сообщение статистики порта используется для получения статистических данных порта коммутатора, определенного в поле "порт" сообщения, а сообщение статистики виртуального канала используется для получения статистических данных для виртуального канала (определенного в полях VPI/VC1 сообщения) на входном порте коммутатора, определенного в поле "порт" сообщения.

Сообщения конфигурации протокола GSMP позволяют контроллеру коммутации определять функциональные возможности ATM коммутатора в базовом коммутационном блоке. Определены три типа сообщений конфигурации протокола GSMP: конфигурация коммутатора, конфигурация порта, конфигурация всех портов. Сообщения конфигурации протокола GSMP используют различные форматы для сообщения запроса и сообщения ответа, поскольку они содержат различную информацию в своих полях. Передаваемое контроллером коммутации к ATM коммутатору, сообщение запроса конфигурации коммутатора, указанное конкретным полем "тип сообщения", запрашивает у ATM коммутатора данные о его глобальной конфигурации. Затем коммутатор возвращает контроллеру коммутации сообщение ответа конфигурации коммутатора, которое включает в себя поля для типа коммутатора и имени коммутатора для ATM коммутатора, а также версию установленного аппаратно-программного обеспечения коммутатора. Тип коммутатора предоставляется изготовителем коммутатора

для идентификации соответствующего изделия, а имя коммутатора может представлять собой 48-битовый адрес IEEE 802 MAC или иную величину, которая является уникальной в операционном контексте коммутатора. Сообщение запроса конфигурации порта имеет свое собственное поле "тип сообщения" и передается контроллером коммутации к ATM коммутатору. Сообщение запроса конфигурации порта запрашивает у коммутатора информацию конфигурации одного порта, который определен в поле "порт" сообщения запроса конфигурации порта. Коммутатор передает контроллеру коммутации сообщение ответа конфигурации порта с индикацией успеха, которое включает в себя информацию конфигурации для входной и выходной сторон определенного порта. Информация конфигурации в сообщении ответа конфигурации порта с индикацией успеха включает следующее: текущий номер сеанса порта, минимальное значение VPI в таблице соединений на входном порте, которое может поддерживаться протоколом GSMP, максимальное значение VPI в таблице соединений на входном порте, которое может поддерживаться протоколом GSMP, минимальное значение VCI в таблице соединений на входном порте, которое может поддерживаться протоколом GSMP, максимальное значение VCI в таблице соединений на входном порте, которое может поддерживаться протоколом GSMP. Информация конфигурации также включает следующее: скорость передачи элементов данных (скорость ATM-элементов данных в секунду) для порта, текущее состояние (т.е. работоспособен, неработоспособен, недоступен, внутреннее закольцовывание, внешнее закольцовывание, двунаправленное закольцовывание) для порта; тип порта (тип физического интерфейса передачи для порта, например неизвестный, SONET STS-3c при 155,52 Мб/с, DS3 при 44,736 Мб/с, 4B/5B кодирование при 100 Мб/с, 8B/10B кодирование при 155,52 Мб/с, физический уровень ATM Forum при 25 Мб/с или физический уровень ATM Forum при 51 Мб/с), и число приоритетов, которое выходной порт может присваивать соединениям виртуального канала. Информация конфигурации обеспечивается со ссылками на запись порта для конкретного порта. Контроллер коммутации передает сообщение запроса конфигурации всех портов, которое имеет собственное поле "тип сообщения", на ATM коммутатор для запроса информации конфигурации для всех коммутационных портов. Таким образом, сообщение запроса конфигурации всех портов специально не определяет конкретный порт. Коммутатор передает сообщение ответа конфигурации всех портов с индикацией успеха, которое обеспечивает следующее: число записей портов, содержащихся в ответном сообщении, длину в байтах каждой записи порта и записи порта для каждого порта. Запись порта для каждого порта представляет собой ту же самую информацию конфигурации, что и описана выше для сообщения ответа конфигурации порта с индикацией успеха. Разумеется, если число записей портов превышает определенный максимум,

установленный для сообщения ответа конфигурации всех портов с индикацией успеха, то записи портов могут быть переданы в нескольких ответных сообщениях с индикацией успеха, которые не превышают установленное максимальное значение.

Сообщения событий протокола GSMP обеспечивают возможность для ATM коммутатора информировать контроллер коммутации об определенных асинхронных событиях. Как упоминалось выше, сообщения событий не подтверждаются. Сообщения событий могут иметь разные типы событий в зависимости от конкретного асинхронного события. Различные сообщения событий включают в себя сообщение "событие работоспособности порта", сообщение "событие неработоспособности порта", сообщение "событие недействительных VPI/VCI", сообщение "событие нового порта", сообщение "событие заблокированного порта". Каждый порт коммутатора поддерживает номер последовательности событий и набор флагов событий (по одному флагу события на каждый тип сообщения). Если порт коммутатора передает сообщение события, он устанавливает соответствующий флаг события на данный порт. Тем самым порту не разрешается передавать сообщение другого события того же самого типа, пока соответствующий флаг события не будет сброшен функцией сброса флагов событий сообщения управления портом протокола GSMP. Использование флагов событий обеспечивает простое управление потоком, препятствуя перегрузке контроллера коммутации сообщениями событий со стороны коммутатора. Номер последовательности события устанавливается на нуль, когда порт инициализирован, и получает приращение всякий раз, когда асинхронное событие, уведомляемое сообщением события, обнаруживается на данном порте, независимо от того, передано ли сообщение события или нет. Текущий номер последовательности события включен в сообщения событий для информирования контроллера коммутации об асинхронных событиях, которые имели место для данного порта, но о которых не было направлено уведомление посредством сообщения события ввиду действия простого механизма управления потоком. Сообщение "событие работоспособности порта" информирует контроллер коммутации, что конкретный порт изменил свое состояние с неработоспособного на работоспособное. Когда порт становится работоспособным, все соединения на его входном порте удаляются (таблицы соединений входного порта являются незаполненными), и коммутатор присваивает новый номер сеанса порта. Сообщение "событие неработоспособности порта" информирует контроллер коммутатора, что конкретный порт изменил свое состояние с работоспособного на неработоспособное. Если коммутатор способен обнаружить отказ линии связи, то коммутатор передает сообщение "событие неработоспособности порта" для уведомления контроллера коммутации о неисправности линии связи. Если один или более элементов ATM данных приходят на входной порт с меткой VPI/VCI, которая в данный момент не присвоена соединению виртуального канала, то

коммутатор передает сообщение "событие недействующего VPI/VCI" к контроллеру коммутации. Сообщение "событие недействующего VPI/VCI" определяет входной порт и VPI/VCI в полях "порт" и "VPI/VCI" соответственно. Сообщение "событие нового порта", определяющее номер нового порта, информирует контроллер коммутации, что к коммутатору добавлен новый порт. Сообщение "событие заблокированного порта" информирует контроллер коммутации, что в коммутаторе удален определенный порт. Сообщение "событие заблокированного порта" определяет номер удаленного порта и номер сеанса порта, который действовал, прежде чем порт был удален, в полях "порт" и "номер сеанса порта" соответственно.

С. Протокол IFMP-C

В соответствии с некоторыми вариантами осуществления системное программное обеспечение может также использовать протокол IFMP-C для установления связи по линии связи между контроллером коммутации базового коммутационного блока (далее называемым IFMP-C контроллером) и исполнительным устройством коммутации (далее называемым IFMP-C исполнительным устройством) и обеспечения при этом распределения пересылки пакетов на уровне 3, если это необходимо. В частности, протокол IFMP-C, представляющий собой универсальный асимметричный протокол для управления IFMP-C исполнительным устройством, выполняется в виртуальном канале, установленном при инициализации в линии связи между IFMP-C контроллером и IFMP-C исполнительным устройством. Простой IFMP-C контроллер может использовать множество конкретизации протокола IFMP-C по отдельным виртуальным каналам для управления множеством IFMP-C исполнительных устройств. Протокол IFMP-C также включает в себя протокол близости IFMP-C, который используется для синхронизации состояний в линии связи между IFMP-C контроллером и IFMP-C исполнительным устройством для обнаружения идентификации узла на другом конце линии и для обнаружения изменений в идентификации этого узла.

Протокол IFMP-C позволяет контроллеру коммутации устанавливать и отменять соединения через ATM коммутатор, добавлять и исключать элементы в соединении от одной точки к множеству точек, управлять интерфейсами IFMP-C, запрашивать информацию конфигурирования и статистические данные. Протокол IFMP-C также позволяет IFMP-C исполнительному устройству информировать IFMP-C контроллер о таких событиях, как отказ в линии связи.

Как упомянуто выше, протокол IFMP-C представляет собой протокол типа "ведущий-подчиненный". IFMP-C контроллер выдает сообщения запроса на IFMP-C исполнительное устройство. Каждое сообщение запроса показывает, требуется ли ответ от IFMP-C исполнительного устройства, и содержит идентификатор операции для обеспечения того, чтобы ответ был связан с конкретным запросом. IFMP-C исполнительное устройство отвечает ответными сообщениями, указывающими

успех или неуспех операции.

Протокол IFMP-C включает протокол близости IFMP-C, которому назначается набор конкретных типов сообщений. Протокол близости IFMP-C используется для установления синхронизации в линии передачи ATM данных и поддерживает квитирование между IFMP-C контроллером и IFMP-C исполнительным устройством. За исключением сообщений протокола близости IFMP-C, никакие другие сообщения IFMP-C не могут передаваться по линии передачи ATM данных до тех пор, пока протокол близости IFMP-C не установит синхронизацию состояний. Все другие сообщения IFMP-C, принятые по линии передачи ATM данных, которые в текущий момент не достигли синхронизации состояний, отбрасываются. В настоящем изобретении протокол IFMP-C имеет четыре класса сообщений: "интерфейс", "ветвь", "управление", "диспетчеризация". Эти четыре класса сообщений являются сообщениями типа "запрос-ответ", причем каждое имеет определенный формат для сообщения запроса и определенный формат для сообщения ответа. Если не указано иное, то сообщение ответа с индикацией сбоя представляет собой то же самое, что и сообщение запроса, которое вызвало этот сбой, за исключением того, что поле "код" указывает характер сбоя. В рассматриваемом варианте осуществления имеется множество других типов сообщений, например сообщения протокола GFMP-C "функции". Каждый из классов сообщений протокола IFMP-C в конкретном варианте осуществления имеет ряд различных типов сообщений.

В настоящем изобретении пакеты протокола IFMP-C имеют переменную длину и инкапсулированы непосредственно в AAL-5 CPCS-PDU с приписанным заголовком LLC/SNAP, аналогично инкапсуляции по умолчанию IP-пакетов в линиях передачи ATM данных, описанных выше со ссылками на фиг.10b. На фиг.15a показан инкапсулированный пакет 1000 протокола IFMP-C. В основном инкапсуляция по умолчанию предусматривает приписывание заголовка LLC/SNAP к пакету протокола IFMP-C, который инкапсулирован в составе полезной нагрузки AAL-5 CPCS-PDU. Инкапсулированный по умолчанию IFMP-C-пакет 1000 содержит заголовок LLC/SNAP (24-битовое поле 1002 LLC, за которым следует 8-битовая часть заголовка 1004 SNAP в первом 32-битовом слове, и оставшаяся 32-битовая часть заголовка 1004 SNAP), IFMP-C сообщение 1006 (которое имеет длину целого кратного 32-битового слова), поле 1008 заполнителя и концовое поле 1010 AAL-5 CPCS-PDU. Поле заполнителя 1008 может иметь длину от 0 до 47 байтов, поле концовика 1010 равно 8 байтам (четыре 32-битовых слова). Передаваемый блок IFMP-C сообщения 1006, использующего инкапсуляцию по умолчанию, равен 1500 байтов. Пакеты, использующие инкапсуляцию по умолчанию, в конкретном варианте осуществления передаются к VPI=0, VCI=15 (установленный по умолчанию виртуальный канал).

На фиг.15b показана структура сообщения 1012 протокола близости IFMP-C, которое

может содержаться в поле 1006 сообщения протокола IFMP-C инкапсулированного пакета 1000 протокола IFMP-C по фиг.15a. Как показано на фиг. 15b, сообщение 1012 протокола близости IFMP-C содержит (в порядке от старшего бита к младшему биту) следующие поля: 8-битовое поле "версия" 1014, 8-битовое поле "тип сообщения" 1016, 8-битовое поле "код" 1018 и 8-битовое поле 1020 "флаг" в качестве первого 32-битового слова; 16-битовое поле 1022 "идентификатор операции" и 16-битовое поле 1024 "длина сообщения" в качестве второго 32-битового слова; поле 1026 "экземпляр синхронизации передатчика" в качестве третьего 32-битового слова; поле 1028 "экземпляр синхронизации равного по положению" в качестве четвертого 32-битового слова и поле 1030 "тело сообщения". При обсуждении сообщений протокола IFMP-C под понятием "передатчик" понимается узел, который передает сообщение протокола IFMP-C по линии связи. Таким узлом может быть IFMP-C контроллер или IFMP-C исполнительное устройство. Обобщенное сообщение 1012 протокола IFMP-C, описанное выше, имеет некоторые различия, как описано ниже, по отношению к сообщениям протокола близости IFMP-C.

При передаче сообщения передатчик устанавливает поле 1014 "версия" в соответствии с номером версии протокола IFMP-C, используемой в данное время. Поле 1016 "тип сообщения" идентифицирует тип и формат поля 1030 "тело сообщения". Каждое конкретное сообщение имеет уникальный тип сообщения в конкретном варианте осуществления. Поле 1018 используется для указания успеха или неудачи при выполнении операции, а поле 1020 "флаг" используется для указания того, каким образом должен обрабатываться пакет и какой тип ответа требуется. Поле 1020 "флаг" может быть установлено как флаг подтверждения выполнения (PLEASE_ACK), когда передатчик требует уведомления от приемника, что операция завершена. Если передатчик требует уведомления только в случае, когда операция неуспешна, то передатчик может устанавливать поле 1020 "флаг" как подтверждение отрицательного результата (PLEASE_NACK). Если операция успешна и сообщение запроса имеет флаг PLEASE_ACK в поле 1020 "флаг", то поле 1020 "флаг" должно быть установлено как флаг подтверждения (ACK), и поле 1018 "код" в сообщении ответа должно быть установлено на предварительно определенное значение, например 0 в конкретном варианте осуществления. Однако если операция неуспешна и сообщение запроса имеет флаг PLEASE_NACK в поле 1020 "флаг", то поле 1020 "флаг" в сообщении ответа должно быть установлено как флаг подтверждения отрицательного результата (NACK), и поле 1018 "код" в сообщении ответа должно быть установлено для указания конкретного случая сбоя. Полный перечень сбоев для использования в поле 1018 "код" рассмотрен ниже. Поле 1022 "идентификатор операции" уникальным образом идентифицирует каждое сообщение. Если сообщение требует ответа, то ответное сообщение имеет то же самое значение в поле 1022 "идентификатор операции", обеспечивая возможность сопоставления сообщений. Поле 1024 "длина

сообщения" указывает длину (включая заголовок протокола IFMP-C, но исключая инкапсуляцию с использованием SNAP/LLC) сообщения в байтах, в соответствии с конкретным вариантом осуществления. Поле 1026 "экземпляр синхронизации передатчика" содержит значение, идентифицирующее текущий экземпляр синхронизации передатчика. Этим значением обмениваются с использованием протокола близости IFMP-C, как описано ниже. После приема сообщения от передатчика значение поля 1026 "экземпляр синхронизации передатчика" сравнивается со значением локального экземпляра равного по положению в приемнике из протокола близости IFMP-C. Если эти экземпляры не совпадают, то пакет игнорируется приемником. Поле 1028 "экземпляр синхронизации равного по положению" содержит значение, идентифицирующее то, что передатчику представляется текущим экземпляром синхронизации его равного по положению. Этим значением обмениваются с использованием протокола близости IFMP-C. После приема сообщения от передатчика значение поля 1028 "экземпляр синхронизации равного по положению" сравнивается со значением локального экземпляра в приемнике из протокола близости IFMP-C. Если эти экземпляры не совпадают, то пакет игнорируется приемником. Поле 1030 "тело сообщения" содержит данные, специфические для сообщения, как описано более детально ниже для различных типов классов сообщений.

На фиг.16a показана структура сообщения 1040 протокола близости IFMP-C, которое может содержаться в поле 1006 сообщения протокола IFMP-C инкапсулированного пакета 1000 протокола IFMP-C по фиг.15a. Как показано на фиг. 16a, сообщение 1040 протокола близости IFMP-C содержит (в порядке от старшего бита к младшему биту) следующие поля: 8-битовое поле 1014 "версия", 8-битовое поле 1016 "тип сообщения", 8-битовое поле 1018 "код", 8-битовое поле 1020 "флаг", поле 1022 "идентификатор операции", поле 1024 "длина сообщения", 32-битовое поле 1042 "экземпляр передатчика", 32-битовое поле 1044 "экземпляр равного по положению", 16-битовое поле 1046 "тип передатчика" и 16-битовое поле 1048 "интервал ACK", поле 1050 "имя передатчика" в виде следующих 48 битов, поле 1052 "имя равного по положению" в виде следующих 48 битов. При обсуждении сообщений протокола близости IFMP-C под понятием "передатчик" понимается узел, который передает сообщение протокола близости IFMP-C, а под понятием "равный по положению" понимается узел, которому передатчик передает сообщение протокола близости IFMP-C по линии связи. Узел может представлять собой IFMP-C контроллер или IFMP-C исполнительное устройство.

В сообщении 1040 протокола близости IFMP-C поле 1014 "версия", поле 1016 "тип сообщения", поле 1018 "код", поле 1020 "флаг", поле 1022 "идентификатор операции" и поле 1024 "длина сообщения" используются так же, как для обобщенного сообщения 1012, описанного выше со ссылками на фиг.15b. Поле 1016 "тип сообщения" для сообщения 1040 протокола близости IFMP-C

устанавливается на конкретное значение для определения конкретного типа сообщения протокола близости IFMP-C. В настоящем варианте осуществления имеются четыре возможных значения для поля 1016 "тип сообщения" в сообщениях протокола близости IFMP-C: SYN (сообщение синхронизации), SYNACK (сообщение подтверждения синхронизации), RSTACK (сообщение подтверждения сброса), ACK (сообщение подтверждения). Поле 1042 "экземпляр передатчика" устанавливается на текущий номер экземпляра передатчика для линии связи. Номер экземпляра используется для обнаружения того, когда одна сторона линии связи перезапускается. Номер экземпляра для линии должен быть уникальным в пределах ближайшего прошлого и уникальным при перезапуске узла. Поле 1044 "экземпляр равного по положению" устанавливается в соответствии с тем, что, как представляется передатчику, является текущим номером экземпляра для удаленной стороны линии связи. Предварительно определенное значение, например 0 в конкретном варианте осуществления, используется в поле 1044 "экземпляр равного по положению" для указания, если в передатчике не известен экземпляр удаленного узла. Это предварительно определенное значение может быть зарезервировано для этой цели и не используется в качестве действительного номера экземпляра в поле 1042 "экземпляр передатчика". Поле 1046 "тип передатчика" указывает тип IFMP-C узла (определенными типами являются IFMP-C контроллер и IFMP-C исполнительное устройство), передающего сообщение. Если узел принимает сообщение протокола близости IFMP-C, то приемник сравнивает поле 1046 "тип передатчика" в принятом сообщении с типом приемника, чтобы предотвратить сопряжение IFMP-C контроллера с другим IFMP-C контроллером или сопряжение IFMP-C исполнительного устройства с другим IFMP-C исполнительным устройством. Если типы одинаковы или являются неопределенными, то сообщение игнорируется. Если типы являются взаимно дополняющими (один из них является IFMP-C контроллером, а другой - IFMP-C исполнительным устройством), то синхронизация осуществляется. Поле 1048 "интервал ACK" устанавливается IFMP-C контроллером для указания предпочтительного интервала подтверждения (ACK). IFMP-C исполнительное устройство использует интервал ACK в качестве интервала таймаута для выполнения протокола близости. При передаче сообщений протокола близости IFMP-C соответствующее исполнительное устройство устанавливает поле 1048 "интервал ACK" на предварительно определенное значение, например 0 в конкретном варианте осуществления, которое игнорируется IFMP-C контроллером. Поле 1050 "имя передатчика" представляет собой значение, уникальное в операционном контексте устройства (например, адрес MAC (контроллера доступа к среде), если он имеется), которое идентифицирует передатчик. Поле 1052 "имя равного по положению" устанавливается передатчиком на значение, которое, как ему

представляется, является именем равного по положению узла на другом конце линии связи. Передатчик устанавливает поле 1052 "имя равного по положению" в предварительно определенное значение, например 0 в конкретном варианте осуществления, для индикации ситуации, когда имя равного по положению узла не известно.

Как обсуждено выше, протокол близости IFMP-C используется для установления синхронизации состояний в линии, соединяющей IFMP-C контроллер и IFMP-C исполнительное устройство, а также для идентификации изменений состояния линии и перезапуска узла на другом конце линии связи. Каждая сторона линии выполняет протокол близости IFMP-C. Для протокола близости IFMP-C существуют три возможных состояния для конкретной линии связи: SYNSENT (сообщение синхронизации послано), SYNRCVD (сообщение синхронизации принято), ESTAB (синхронизация установлена). Синхронизация состояний в линии (при установке взаимодействия с соседним узлом интерфейсы должны быть в состоянии ESTAB) требуется, прежде чем IFMP-C контроллер и IFMP-C исполнительное устройство смогут передавать сообщения протокола IFMP-C.

Для протокола близости IFMP-C существуют два типа событий, которые могут вызвать изменение состояний: события, вызванные таймером, и приход пакета. Эти изменения состояний иллюстрируются на фиг.16b, на которой показана диаграмма состояний, иллюстрирующая работу передающего узла (IFMP-C контроллера или IFMP-C исполнительного устройства) в трех возможных состояниях протокола близости IFMP-C. IFMP-C контроллер устанавливает интервал таймера в IFMP-C исполнительном устройстве путем установки поля 1048 "интервал ACK" на конкретное значение, например 1 с, в конкретном варианте осуществления. Разумеется, в других вариантах осуществления могут быть использованы другие интервалы таймера. В каждом узле необходим таймер для периодического генерирования сообщений протокола близости IFMP-C SYN, SYNACK, ACK, как описано ниже.

События, вызванные таймером, для протокола близости IFMP-C рассмотрены ниже. Как показано на фиг.16b, по истечении установленного времени таймера (обозначено как t), когда передающий узел находится в состоянии 1060 SYNSENT, передающий узел сбрасывает таймер и посылает сообщение протокола близости IFMP-C SYN (показано как 1062). Это действие (показано пунктирной линией) выполняется только IFMP-C исполнительным устройством, которое должно начать процедуру синхронизации путем передачи пакета SYN в соответствии с конкретным вариантом осуществления. По истечении времени таймера, когда передающий узел является IFMP-C контроллером, узел просто сбрасывает таймер, без передачи пакета SYN. По истечении установленного времени таймера, когда передающий узел находится в состоянии 1064 SYNRCVD, передающий узел сбрасывает таймер и посылает сообщение протокола близости IFMP-C SYNACK

(показано как 1066). По истечении установленного времени таймера, когда передающий узел находится в состоянии 1068 ESTAB, передающий узел сбрасывает таймер и посылает сообщение протокола близости IFMP-C ACK (показано как 1070). Для блокировки по времени протокола близости IFMP-C обе стороны линии связи должны установить линию связи в исходное состояние и перейти в состояние SYNSENT, если они проходят через предварительно определенное число (например, три) периодов блокировки по времени без приема пакета ACK от другой стороны линии связи. Когда IFMP-C исполнительное устройство блокируется по времени, оно должно восстановить интервал ACK в его значение, установленное по умолчанию.

Помимо обсужденных выше переходов, обусловленных таймером, переходы состояний в протоколе IFMP-C также вызываются приходом пакета протокола близости IFMP-C. Когда сообщение протокола близости IFMP-C поступает в узел, то осуществляется определенное действие на основе текущего состояния узла, содержимого сообщения и типа сообщения. В соответствии с протоколом близости IFMP-C осуществляются следующие операции: операция обновления равного по положению и операция сброса линии. Каждое событие в протоколе близости IFMP-C сохраняет состояние, которое определяет экземпляр равного по положению и его имя (т.е. экземпляр и имя равного по положению узла на удаленном конце линии связи). Операция обновления равного по положению устанавливает локальное состояние равного по положению, совпадающим с состоянием передатчика из самого последнего принятого сообщения протокола близости IFMP-C. При перезапуске синхронизации операция сброса линии связи генерирует новый номер экземпляра для локальной стороны линии связи и удаляет состояние равного по положению путем установки экземпляра и имени равного по положению в нуль.

Для последующего описания со ссылками на фиг.16b условие "%X" определяется следующим образом: поля "экземпляр равного по положению" и "имя равного по положению" во входящем сообщении совпадают с локальными значениями экземпляра и имени, связанными с линией связи. Условие "%Y" определяется следующим образом: поля "экземпляр передатчика" и "имя передатчика" во входящем сообщении совпадают со значениями экземпляра передатчика и имени передатчика, запомненными для экземпляра равного по положению и имени равного по положению. На фиг.16b условие "A" означает, что передающий узел принимает входящее сообщение SYNACK протокола близости IFMP-C, и что условие "%X" удовлетворено. Условие "B" означает, что передающий узел принимает входящее сообщение SYNACK протокола близости IFMP-C, и что условия "%X" и "%Y" удовлетворены. Условие "D" означает, что передающий узел принимает входящее сообщение ACK протокола близости IFMP-C,

и что условия "%X" и "%Y" не удовлетворены. Условие "E" означает, что передающий узел принимает входящее сообщение RSTACK протокола близости IFMP-C, и что условия "%X" и "%Y" удовлетворены. Условие "F" означает, что передающий узел принимает входящее сообщение RSTACK протокола близости IFMP-C, и что условия "%X" и "%Y" не удовлетворены. Условие "G" означает, что передающий узел принимает сообщение SYN или сообщение SYNACK протокола близости IFMP-C, и что условие "%X" удовлетворено. Условие "H" означает, что передающий узел принимает сообщение SYN или сообщение SYNACK протокола близости IFMP-C, и что условие "%X" не удовлетворено.

Возможны различные переходы, когда узел находится в состоянии 1060 SYNSENT. Как показано на фиг.16b, если передающий узел находится в состоянии 1060 SYNSENT и принимает входящее сообщение SYN протокола близости IFMP-C от равного по положению на другом конце линии связи, то передающий узел выполняет операцию обновления равного по положению и передает сообщение SYNACK протокола близости IFMP-C к равному по положению (показано этапом 1072). Затем передатчик переходит из состояния SYNSENT 1060 в состояние SYNRCVD 1064. Если передатчик принимает входящее сообщение RSTACK протокола близости IFMP-C, находясь в состоянии SYNSENT 1060, то передающий узел продолжает находиться в состоянии SYNSENT 1060, но пропускает сообщение (показано как 1074). Если передатчик принимает входящее сообщение ACK протокола близости IFMP-C или удовлетворено условие B, то передающий узел продолжает находиться в состоянии SYNSENT 1060, но передает сообщение RSTACK протокола близости IFMP-C (показано как 1076). Если условие A удовлетворено в состоянии SYNSENT 1060, то передающий узел выполняет операцию обновления равного по положению, передает сообщение ACK протокола близости IFMP-C (показано как 1078) и переходит в состояние ESTAB 1068.

Если узел находится в состоянии SYNRCVD 1064, то возможны различные переходы. Как показано на фиг.16b, если передающий узел находится в состоянии SYNRCVD 1064 и принимает входящее сообщение SYN протокола близости IFMP-C от равного по положению на другом конце линии связи, то передающий узел выполняет операцию обновления равного по положению и передает сообщение SYNACK протокола близости IFMP-C к равному по положению (показано этапом 1072), оставаясь в состоянии SYNRCVD 1064. Если условие B или D удовлетворено в состоянии SYNRCVD 1064, то передающий узел остается в состоянии SYNRCVD 1064 и передает сообщение RSTACK протокола близости IFMP-C (показано как 1078). Если условие F удовлетворено, когда узел находится в состоянии SYNRCVD 1064, то узел остается в состоянии SYNRCVD 1064, но пропускает сообщение (показано как 1080). Если условие E удовлетворено, когда узел находится в состоянии SYNRCVD 1064, то передающий узел выполняет операцию сброса линии, передает сообщение SYN протокола близости

IFMP-C (показано как 1082) и переходит из состояния SYNRCVD 1064 в состояние SYNSENT 1060. Если условие A удовлетворено, когда узел находится в состоянии SYNRCVD 1064, то передающий узел выполняет операцию обновления равного по положению, передает сообщение ACK протокола близости IFMP-C (показано как 1078) и переходит из состояния SYNRCVD 1064 в состояние ESTAB 1068. Если условие C удовлетворено, когда узел находится в состоянии SYNRCVD 1064, то передающий узел передает сообщение ACK протокола близости IFMP-C (показано как 1084) и переходит из состояния SYNRCVD 1064 в состояние ESTAB 1068.

Если узел находится в состоянии ESTAB 1068, то возможны различные переходы. Как показано на фиг.16b, если условие D или H удовлетворено при нахождении в состоянии ESTAB 1068, передающий узел остается в состоянии ESTAB 1068 и передает сообщение RSTACK протокола близости IFMP-C (показано как 1088). Если условие F удовлетворено, когда узел находится в состоянии ESTAB 1068, то узел остается в состоянии ESTAB 1068, но пропускает сообщение (показано как 1080). Если условие E удовлетворено, когда узел находится в состоянии ESTAB 1068, то передающий узел выполняет операцию сброса линии и передает сообщение SYN протокола близости IFMP-C (показано как 1086) и переходит из состояния ESTAB 1068 в состояние SYNSENT 1060. Если состояние S или G удовлетворено при нахождении узла в состоянии ESTAB 1068, то передающий узел передает сообщение ACK протокола близости IFMP-C (показано как 1090) и остается в состоянии ESTAB 1068. Каждый узел на одной стороне линии связи не должен передавать более одного сообщения ACK протокола близости IFMP-C, генерируемого приходом пакета, на период таймаута в соответствии с конкретным вариантом осуществления.

Сообщение интерфейса IFMP-C может содержаться в поле "сообщение" 1006 протокола IFMP-C инкапсулированного пакета 1000 протокола IFMP-C, показанного на фиг.15a. Как упомянуто выше, после того как синхронизация линии установлена, сообщения интерфейса IFMP-C используются для обнаружения и конфигурирования интерфейсов IFMP-C исполнительного устройства. В конкретном варианте осуществления сообщения интерфейса IFMP-C включают сообщения запроса перечня интерфейсов и соответствующего ответа, сообщения ошибок перечня интерфейсов, сообщения обращения с запросом к интерфейсу и соответствующего ответа, сообщения ошибок запроса к интерфейсу, сообщения запроса и ответа конфигурирования интерфейса, сообщения ошибок конфигурирования интерфейса. Сообщения перечня интерфейсов и сообщения запроса к интерфейсу протокола IFMP-C имеют специфические значения в поле 1016 "тип сообщения". Разумеется, в других вариантах осуществления возможны дополнительные сообщения или сообщения, выполняющие сходные функции.

Сообщения запроса перечня интерфейсов и соответствующего ответа и сообщения ошибок перечня интерфейсов используются

для определения того, какие интерфейсы доступны на IFMP-C исполнительном устройстве. На фиг.17a и 17b представлены структуры сообщений запроса и ответа перечня интерфейсов соответственно. Как показано на фиг.17a, сообщение запроса перечня интерфейсов 1100 имеет обобщенный формат, который был рассмотрен выше со ссылками на фиг. 15b, причем поле 1030 "тело сообщения" содержит поле 1112 "следующее "печенье"" (идентификатор, используемый для координации в межадачном взаимодействии), представляющее собой 32-битовое значение, возвращаемое из предыдущего ответного сообщения перечня интерфейсов. Поле 1112 "следующий идентификатор" используется для поддержки перечней идентификаторов, которые занимают более одного сообщения. Значение поля 1112 "следующий идентификатор" представляет собой значение, возвращаемое IFMP-C исполнительным устройством, для обеспечения возможности следующему сообщению запроса осуществлять продолжение перечня интерфейсов из предыдущего завершенного сообщения. Предварительно определенное значение, например, в конкретном варианте осуществления, используется для указания того, что перечень интерфейсов должен запускаться с начала. Если сообщение запроса перечня интерфейсов имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаги" и операция успешна, то IFMP-C исполнительное устройство возвращает сообщение ответа 1114 перечня интерфейсов (фиг.17b) с его полем 1020 "флаги", установленным на флаг ACK, и с полем 1018 "код", установленным на предварительно определенное значение (например, 0 в конкретном варианте осуществления), указывающее отсутствие ошибок. Как показано на фиг.17b, поле 1112 "следующий идентификатор" в сообщении ответа 1114 перечня интерфейсов представляет собой 32-битовое значение, возвращаемое как часть сообщения. Если значение поля 1112 "следующий идентификатор" равно 0, то все интерфейсы перечислены. Если значение не равно 0, то это значение используется в качестве аргумента для следующего сообщения перечня интерфейсов для получения имеющихся остальных интерфейсов. IFMP-C исполнительное устройство присваивает каждому интерфейсу уникальный 32-битовый идентификатор, который используется в других сообщениях протокола IFMP-C для ссылки на конкретный интерфейс. Ответное сообщение 1114 перечня интерфейсов перечисляет в поле 1118 "идентификатор 1 интерфейса", в поле 1120 "идентификатор 2 интерфейса" и т.д. все идентификаторы для каждого интерфейса на IFMP-C исполнительном устройстве, которые могут быть перечислены в ответном сообщении.

Если запрос перечня сообщений был неуспешным, то IFMP-C исполнительное устройство возвращает сообщение ошибок перечня интерфейсов, которое содержит заголовок протокола IFMP-C. В сообщении ошибок перечня интерфейсов флаг NACK должен быть установлен в поле 1020 "флаги", поле 1018 "код" должно указывать причину

сбоя, а поле 1022 "идентификатор операции" должно быть тем же самым, что и в сообщении запроса перечня интерфейсов. В числе причин сбоев, каждая из которых имеет конкретное значение для использования в поле 1018 "код", может быть то, что в поле 1112 был указан недействительный идентификатор, IFMP-C исполнительное устройство не смогло назначить буфер сообщения для завершения ответа, специфическая ошибка пользователя обусловила невозможность завершения запроса и т.д.

Сообщения обращения с запросом к интерфейсу и соответствующего ответа и сообщения ошибок запроса к интерфейсу используются для получения атрибутов, связанных с определенным интерфейсом на IFMP-C исполнительном устройстве. Фиг. 17c и 17d показывают структуру сообщения обращения с запросом к интерфейсу и соответствующего ответа. Как видно из фиг.17c, сообщение 1130 обращения с запросом к интерфейсу имеет обобщенный формат, как ранее описано со ссылками на фиг. 15b для поля 1030 "тело сообщения", содержащего поле 1132 "идентификатор интерфейса", которое представляет собой 32-битовый идентификатор, уникальным образом идентифицирующий интерфейс, атрибуты которого запрашиваются. Как описано выше, идентификаторы интерфейсов присваиваются IFMP-C исполнительным устройством и могут быть получены с помощью сообщений перечня интерфейсов. Если сообщение 1130 обращения с запросом к интерфейсу имеет флаг PLEASE_ACK в его поле 1020 "флаги" и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает в ответ сообщение 1134 ответа на запрос к интерфейсу (показано на фиг.17d), в котором в поле 1020 "флаги" установлен флаг ACK, а поле 1018 "код" установлено на предварительно определенное значение, указывающее отсутствие ошибки. Как показано на фиг. 17d, поле 1030 "тело сообщения" в сообщении 1134 ответа на запрос к интерфейсу также содержит 48-битовое поле 1136 "имя интерфейса" предназначенное для указания имени запрашиваемого интерфейса (в конкретном варианте осуществления может быть использован MAC адрес), 8-битовое поле 1138 "тип интерфейса" (значения могут быть определены для различных типов интерфейсов, таких как ATM, Ethernet, FastEthernet, Gigabit Ethernet, FDDI или другие типы интерфейсов локальных сетей), 8-битовое поле 1140 "тип среды передачи" (указывающий физическую среду интерфейса, например многомодовое волокно, скрученная пара категории 5, одномодовое волокно и т.п.). Кроме того, сообщение 1134 ответа на запрос к интерфейсу также содержит 32-битовое поле 1142 "поддерживаемые скорости" и 32-битовое поле 1144 "текущая скорость". Поле 1142 "поддерживаемые скорости" указывает различные скорости передачи (например, 10, 25, 100, 155, 622, 1000 Мбит/с и другие), которые поддерживает запрашиваемый интерфейс. Для интерфейсов, поддерживающих более одной скорости передачи, различные флаги могут быть установлены в поле 1142. Флаг

автоматического согласования может быть установлен в поле 1142 "поддерживаемые скорости", если интерфейс поддерживает автоматическое согласование установок скоростей. Поле 1144 "текущая скорость" указывает текущую скорость передачи для запрашиваемого интерфейса. Если интерфейс находится в режиме автоконфигурирования, то текущая скорость интерфейса указывается в поле 1144 "текущая скорость", и флаг автоматического согласования установлен в поле 1142 "поддерживаемые скорости". Кроме того, сообщение 1134 ответа на запрос к интерфейсу также содержит 32-битовое поле 1146 "поддерживаемый дуплексный режим" (указывающее скорости дуплексного режима, поддерживаемые запрашиваемым интерфейсом, в том числе в полудуплексном режиме, полностью дуплексном режиме или режим автоматического согласования при установке дуплексного режима; более одного флага может быть установлено для интерфейсов, поддерживающих более одной дуплексной установки), и 32-битовое поле 1148 "текущий дуплексный режим" (указывающее текущие установки дуплексного режима интерфейса; если интерфейс находится в режиме автоматического согласования, поле 1146 будет указывать соответствующее значение). Кроме того, сообщение 1134 ответа на запрос к интерфейсу также содержит 32-битовое поле 1150 "идентификатор слота интерфейса" (указывает физический слот, который занимает интерфейс на IFMP-C исполнительном устройстве), 32-битовое поле 1152 "идентификатор порта интерфейса" (идентифицирует физический порт, который занимает интерфейс на IFMP-C исполнительном устройстве), 16-битовое поле 1154 "флаги интерфейса" и 16-битовое поле 1156 "статус интерфейса". Поле 1154 "флаги интерфейса" определяет текущие установки опций конфигурации на запрашиваемом интерфейсе, причем каждый флаг указывает различное состояние (например, интерфейс приведен в работоспособное состояние, интерфейс находится в смешанном режиме, принимает все групповые пакеты и т.п.). Поле 1156 "статус интерфейса" указывает текущую информацию статуса (например, трафик управления протокола IFMP-C реализуется через данный интерфейс или канал на этом интерфейсе, состояние работоспособности и т.п.) в связи с линией, которая не подвергается изменениям посредством IFMP-C контроллера). Кроме того, сообщение 1134 ответа на запрос к интерфейсу также содержит 32-битовое поле 1158 "минимальная метка приема" и 32-битовое поле 1160 "максимальная метка приема", 32-битовое поле 1162 "минимальная метка передачи" и 32-битовое поле 1164 "максимальная метка передачи". Если интерфейс является ATM интерфейсом, то поле 1158 "минимальная метка приема" и 1160 "максимальная метка приема" соответственно указывают минимальный и максимальный идентификатор виртуального канала, по которому интерфейс может осуществлять прием. Если интерфейс является ATM интерфейсом, то поле 1162 "минимальная метка передачи" и 1164 "максимальная метка передачи"

соответственно указывают минимальный и максимальный идентификатор виртуального канала, по которому интерфейс может осуществлять передачу. Если запрашиваемый интерфейс является интерфейсом иного типа, то поля 1158, 1160, 1162 и 1164 устанавливаются в нуль.

Если обращение с запросом к интерфейсу безуспешно, то IFMP-C исполнительное устройство передает сообщение ошибок запроса интерфейса, которое содержит IFMP-C заголовок, имеющий флаг NACK, установленный в поле 1020 "флаги", соответствующую причину сбоя, указанную полем 1018 "код", причем поле 1022 "идентификатор операции" то же самое, что и в сообщении 1130 обращения с запросом к интерфейсу. Возможные причины сбоя, которые могут быть указаны в поле 1018 "код" сообщения ошибок запроса интерфейса, включают следующее: недействительный идентификатор интерфейса, неспособность IFMP-C исполнительного устройства назначить буфер сообщения для завершения ответа, специфическая пользовательская ошибка, препятствующая завершению ответа и т. п.

Сообщения запроса и ответа конфигурации интерфейса и сообщения ошибок конфигурации используются для обеспечения возможности IFMP-C контроллеру изменить конфигурацию интерфейса на IFMP-C в исполнительном устройстве. На фиг. 17е показана структура сообщения 1170 запроса конфигурации интерфейса. Как показано на фиг. 17е, сообщение 1170 запроса конфигурации интерфейса имеет обобщенный формат, как ранее описано со ссылками на фиг. 15 b, причем поле 1030 "тело сообщения" содержит поле 1132 "идентификатор интерфейса" (уникальным образом идентифицирующее интерфейс, конфигурация которого изменяется), 16-битовое поле 1172 "очистка флагов", 16-битовое поле 1174 "установка флагов", 32-битовое поле 1176 "скорость", 32-битовое поле 1178 "дуплексный режим". Поле 1172 "очистка флагов" указывает, какие флаги IFMP-C исполнительное устройство должно очистить на определенном интерфейсе, чтобы изменить операцию интерфейса. Примерами очистки флагов могут служить следующие: перевод интерфейса в неработоспособное состояние, вывод интерфейса из смешанного режима, предотвращение приема интерфейсом всех пакетов групповой передачи и т. д. Поле 1174 "установка флагов" указывает, какие флаги IFMP-C исполнительное устройство должно установить на определенном интерфейсе, чтобы изменить операцию интерфейса. Примерами установки флагов могут служить следующие: перевод интерфейса в работоспособное состояние, ввод интерфейса смешанный режим, обеспечение возможности приема интерфейсом всех пакетов групповой передачи и т. д. Поле 1176 "скорость" используется для обеспечения возможности IFMP-C контроллеру изменить установку скорости (из числа поддерживаемых интерфейсом скоростей, а также автоматического согласования, 10, 25, 100, 155, 622, 1000 Мбит/с и другие) определенного интерфейса. Поле 1178 "дуплексный режим" используется для

обеспечения возможности IFMP-C контроллеру изменить установку дуплексного режима (из числа поддерживаемых интерфейсом установок дуплексного режима, в том числе автоматически согласованного дуплексного, полудуплексного или полностью дуплексного режима) определенного интерфейса. Установка поля 1176 "скорость" или поле 1178 "дуплексный режим" на предварительно определенное значение, например, на нуль в конкретном варианте осуществления, указывает, что установка скорости или дуплексного режима не может быть изменена. Если сообщение 1170 запроса конфигурации интерфейса от IFMP-C контроллера имеет флаг PLEASE_ACK в его поле 1020 "флаги" и операция успешно осуществлена, то IFMP-C исполнительное устройство возвращает в ответ сообщение ответа конфигурации интерфейса, состоящее из полей IFMP-C заголовка (все поля, показанные на фиг. 15b, кроме поля "тело сообщения") с флагом ACK, установленным в его поле 1020 "флаги", и полем 1018 "код", установленным на предварительно определенное значение, указывающее отсутствие ошибки.

Если сообщение запроса конфигурации интерфейса безуспешно, то IFMP-C исполнительное устройство возвращает в ответ сообщение ошибок конфигурации интерфейса, состоящее из IFMP-C заголовка. В сообщении ошибок перечня интерфейсов флаг NACK должен быть установлен в поле 1020 "флаги", поле 1018 "код" должно указывать причину сбоя, а поле 1022 "идентификатор операции" должно быть тем же самым, что и в сообщении запроса перечня интерфейсов. Возможные причины сбоя, которые могут быть указаны в поле 1018 "код", включают следующее: недействительный идентификатор интерфейса, использование недействительных параметров конфигурации, специфическая пользовательская ошибка, препятствующая завершению ответа и т. п.

Помимо сообщений протокола близости IFMC-C и интерфейса, другие типы сообщений 1012 протокола IFMP-C включают сообщения запроса перехода, соответствующее сообщение ответа и сообщение ошибок. Существуют шесть типов сообщений переходов протокола IFMP-C, используемых для создания, модифицирования, исключения ветвей для пересылки. Более конкретно упомянутые сообщения переходов протокола IFMP-C включают следующие: "добавить переход", "удалить переход", "удалить дерево", "переместить переход", "считать переход", "получить статистику дерева". Каждая из ветвей пересылки имеет значение предпочтения. Если имеет место множество совпадений, то используется ветвь с самым низким предпочтением.

Как упомянуто выше, основная пересылка осуществляется с помощью ветвей, причем каждая ветвь состоит из двух компонентов: входные данные и выходные данные. Входные данные обеспечивают достаточную информацию для обеспечения того, чтобы входящие пакеты могли быть согласованы с ветвями, а выходные данные обеспечивают информацию, необходимую для пересылки пакета, для которого установлено совпадение входных данных. Основные операции,

используемые для модифицирования состояния пересылки IFMP-C исполнительного устройства, включают "добавить переход" и "удалить переход". Добавление более одного перехода (ветви) при одних и тех же входных данных приводит к получению дерева, которое может быть использовано для пересылки каждого входящего пакета к множеству адресатов. Например, первое сообщение "добавить переход", имеющее конкретные входные данные и некоторые выходные данные, устанавливает однократную пересылку пакета. Второе сообщение "добавить переход", имеющее те же самые входные данные и другие выходные данные, преобразует однократную пересылку в многократную пересылку путем добавления другой выходной ветви, связанной с теми же самыми входными данными. Другие выходные ветви могут быть добавлены тем же самым способом с использованием последующих сообщений "добавить переход". Сообщение "удалить переход" для многократного соединения с двумя ветвями удаляет ветвь, имеющую определенные выходные данные, преобразуя многократную пересылку в однократную пересылку. Еще одно сообщение протокола IFMP-C, сообщение "удалить дерево", используется для удаления всех ветвей, совместно использующих одни и те же входные данные. Другое сообщение протокола IFMP-C, сообщение "переместить переход", используется для изменения существующей выходной информации для существующей ветви. Дополнительные сообщения протокола IFMP-C, сообщение "получить статистику дерева", используемое для получения статистики для передающих узлов, входные данные которых определены, и сообщения "считать переход", используемые для целей диагностики и отладки, позволяют IFMP-C контроллеру получать данные обо всех ветвях пересылки для данного IFMP-C исполнительного устройства.

Сообщение "добавить переход" и сообщения запроса "удалить переход" представляют собой сообщения протокола IFMP-C, которые используют один и тот же формат 1200 сообщения (но имеют разные типы сообщений), показанный на фиг. 18а. Как показано на фиг. 18а, формат 1200 сообщения запроса "добавить/удалить переход" протокола IFMP-C имеет обобщенный формат, как описано выше со ссылками на фиг. 15b с полем 1030 "тело сообщения", включающим следующие поля: поле 1201 "идентификатор входного интерфейса", 16-битовое поле 1202 "входное предшествование", 16-битовое поле 1204 "входные флаги", 32-битовое поле 1206 "идентификатор выходного интерфейса", 24-битовое резервное поле 1208, 8-битовое поле 1210 "длина ключа", 8-битовое поле 1212 "длина входного заголовка", 8-битовое поле 1214 "длина удаления", 8-битовое поле 1216 "тип преобразования", 8-битовое поле "длина данных преобразования", 32-битовое поле 1222 "обработка качества обслуживания", поле 1224 "данные ключа" предварительно определенной длины, поле 1225 "маска входного ключа" предварительно определенной длины, поле 1228 "выходные данные заголовка" предварительно

определенной длины и поле 1230 "данные преобразования".

Поля, представленные на фиг. 18а (иные, чем показанные на фиг. 15b поля обобщенного заголовка протокола IFMP-C), описаны ниже. Присвоенное IFMP-C исполнительным устройством и полученное с помощью сообщений перечня интерфейсов протокола IFMP-C поле "идентификатор входного интерфейса" 1201 уникальным образом идентифицирует конкретный входной интерфейс, к которому относится входная ветвь. Поле 1202 "входное предшествование", в конкретном варианте осуществления представляющее собой 16-битовое число без знака, обозначает предшествование, присвоенное ветви. При сравнении входящих пакетов с входными ключами ключ с самым низким предшествованием сравнивается первым. Если более одного элемента имеют одинаковое предшествование, то IFMP-C исполнительное устройство может выбрать любую из сравниваемых ветвей для пересылки пакета. Поле 1204 "входные флаги", используемое для входной ветви, и флаги, указанные в этом поле, используются для обозначения конкретного действия, которое должно быть предпринято, если пакеты согласованы с этим передающим узлом. Примерами такого действия, обозначенными такими флагами, могут служить следующие: "передача управления вниз" - поиск на следующем уровне предшествования после передачи пакета, вместо завершения; "отбрасывание" - игнорирование всех пакетов, которые согласованы с этим входным узлом. Поле 1206 "идентификатор выходного интерфейса" уникальным образом идентифицирует интерфейс, предназначенный для использования для передачи пакета. Резервное поле 1208, которое может быть зарезервировано для использования в будущем, в конкретном варианте осуществления может быть установлено передающим узлом на нуль, и игнорируется приемным узлом, если это поле не используется. Поле 1210 "длина ключа", представляющее собой в конкретном варианте осуществления 8-битовое целое число без знака, определяет длину поля 1224 "данные входного ключа" и поля 1226 "маска входного ключа" в байтах. Поле 1212 "длина входного заголовка", представляющее собой в конкретном варианте осуществления 8-битовое целое число без знака, определяет длину поля 1228 "данные выходного заголовка" в байтах. Поле 1214 "длина удаления", представляющее собой 8-битовое целое число без знака, определяет число байтов для удаления от начала пакета, прежде чем применять преобразование, указанное в поле 1230 "данные преобразования".

Поле 1216 "тип преобразования" определяет тип модифицирования (например, не модифицировать, преобразовать IP- пакет в IFMP- пакет типа 1, IP-пакет в IFMP- пакет типа 2, IFMP- пакет типа 1 в IP- пакет, IFMP- пакет типа 2 в IP-пакет, стандартная IP-пересылка, усечение пакета или иные изменения), которое должно быть осуществлено перед передачей пакета. Некоторые типы преобразований требуют использования данных, которые не являются

частью пакета, так что эти требуемые данные предусматриваются в поле 1230 "данные преобразования". Более конкретно преобразование типа "IP- пакет в IFMP- пакет типа 1 (или 2)" обеспечивает преобразование из протокола IPv4 (или иной версии IP пакета) в инкапсуляцию для потока IFMP протокола типа 1 (или 2), как рассмотрено выше. Преобразование типа "IFMP- пакет типа 1 (или 2) в IPv4- пакет" обеспечивает преобразование пакета, приходящего как IFMP- пакет типа 1 (или 2) в IPv4- пакет с использованием поля 1230 "данные преобразования" для восстановления удаленных полей, а также обновления контрольной суммы TTL и IP-заголовка. Тип преобразования "стандартная IP-пересылка" уменьшает TTL в IP-заголовке и обновляет контрольную сумму IP заголовка. Тип преобразования "усечение пакета" ограничивает размер пакета до величины, определенной в поле 1230 "данные преобразования", передаваемыми данными.

Как показано на фиг. 18а, поле 1218 "длина данных преобразования" представляет собой 8-битовое целое число, определяющее длину поля 1230 "данные преобразования", которое включено в формат 1200 сообщения "добавить/удалить переход". Если преобразование не требует дополнительных данных в поле 1230 "данные преобразования", то поле 1218 "длина данных преобразования" установлено на нуль. Поле 1222 "обработка качества обслуживания" указывает, каким образом пакеты, согласованные с ветвью, должны обрабатываться IFMP-C исполнительным устройством. Поле 1224 "входные данные ключа", длина которого определяет полем 1210 "длина ключа", содержит данные, которые сравниваются с входящим пакетом для установления совпадения данных для узла пересылки. Данные в поле 1224 "данные ключа" включают в себя информацию уровня линии связи, в том числе MAC-адрес, информацию уровня 3, в том числе GP-адрес и т.д. Поле 1226 "маска ключа", длина которого определяется полем 1210 "длина ключа", используется для определения соответствующих битов данных ключа при сравнении информации в поле 1224 "входные данные ключа" с входящим пакетом. Поле 1228 "выходные данные заголовок", длина которого определяется полем 1212 "длина выходного заголовка", содержит заголовок (в типовом случае информацию на уровне линии связи, такую как MAC-заголовки на интерфейсе Ethernet, или VPI/VCI на ATM интерфейсах), которая должна быть присоединена к пакету, прежде чем он будет передан. Как упоминалось выше, поле 1230 "данные преобразования" содержат данные, необходимые для выполнения конкретного преобразования. Для преобразования типа "IFMP- пакет типа 1 (или 2) в IPv4- пакет" данные в поле 1230 "данные преобразования" аналогичны идентификаторам потока, показанным на фиг. 7а (или на фиг. 7b для типа 2 потока, за исключением того, что поля "тип обслуживания" и "протокол" могут быть зарезервированы в некоторых вариантах осуществления). Если преобразование "усечение пакета" определено так, что конкретная копия пакета передается конкретному адресату, то данные 1240 в поле

1230 "данные преобразования" включают в себя 16-битовое резервное поле 1242, которое может быть установлено в нуль передатчиком, и будет игнорироваться принимающим узлом, и 16-битовое поле 1244 "длина усечения", как показано на фиг. 18b. Поле 1244 "длина усечения" указывает, какое количество байтов пакета должно быть передано с интерфейса. Если приходящий пакет длиннее, чем это число, то он сокращается до этой длины. Такая длина усечения представляет собой число байтов, получаемое после удаления некоторого числа байтов, и не включает в себя выходного заголовка, который может быть дополнительно введен.

Если сообщение запроса "добавить/удалить переход" протокола IFMP-C имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаги", и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает ответное сообщение 1250 "добавить/удалить переход" протокола IFMP-C с форматом, показанным на фиг. 18с, имеющее флаг ACK, установленный в его поле 1020 "флаги", с полем 1018 "код", установленным на предварительное определенное значение, указывающее отсутствие ошибки. Как показано на фиг. 18с, ответное сообщение имеет поля заголовка, присущие обобщенному сообщению протокола IFMP-C, причем тело сообщения содержит 16-битовое резервное поле 1252 и 16-битовое поле 1254 "выходной отсчет". Резервное поле 1252 зарезервировано для будущего использования, может быть установлено передатчиком в нуль и при этом игнорируется приемником. Поле 1254 "выходной отсчет", представляющее собой целое число без знака, содержит число выходных ветвей, которые совместно используют одну и ту же входную информацию, после применения текущей операции (если отсчет ветвей равен 1, то имеет место ветвь однократной передачи; если отсчет ветвей более 1, то имеет место ветвь многократной передачи). Такое поле 1254 используется IFMP-C контроллером для обеспечения проверки соответствия состояния ветвей после каждой операции.

Если сообщение запроса "добавить/удалить переход" протокола IFMP-C было неуспешным, то IFMP-C исполнительное устройство передает сообщение ошибок "добавить/удалить переход" протокола IFMP-C. Сообщение ошибок содержит IFMP-C заголовок, в котором содержится то же самое поле 1022 "идентификатор операции", что и в соответствующем сообщении запроса, в поле "флаги" установлен флаг NACK, а в поле 1018 "код" установлен на значение, указывающее причину ошибки. В качестве причин сбоев можно указать следующие: один из идентификаторов интерфейсов недействителен; входная длина ключа больше, чем максимальное значение, поддерживаемое IFMP-C исполнительным устройством; выходное преобразование не поддерживается или не распознается IFMP-C исполнительным устройством; IFMP-C исполнительное устройство имеет недостаточно ресурсов для завершения запроса; имеется другая ветвь с тем же

самым входным ключом, но с другими флагами; параметры качества обслуживания недействительны или не поддерживаются IFMP-C исполнительным устройством; входной ключ или входная маска не поддерживаются IFMP-C исполнительным устройством; конкретная ветвь, которую необходимо добавить, уже существует в IFMP-C исполнительном устройстве; конкретная ветвь, которую необходимо удалить, не существует в IFMP-C исполнительном устройстве; конкретная пользовательская ошибка препятствует завершению запроса или другие причины.

Как показано на фиг.18d, сообщение 1260 запроса "удалить дерево" протокола IFMP-C имеет формат сообщения, содержащий многие из полей, описанных со ссылками на фиг.18a. Формат сообщения 1260 запроса "удалить дерево" представляет собой обобщенный формат, ранее описанный со ссылками на фиг.15b, причем поле 1030 "тело сообщения" включает следующие поля: поле 1201 "идентификатор входного интерфейса", поле 1202 "входное предшествование", поле 1204 "входные флаги", 56-битовое резервное поле, поле 1210 "длина ключа", поле 1224 "входные данные ключа" предварительно определенной длины и поле 1226 "маска входного ключа" предварительно определенной длины.

Поле 1201 "идентификатор входного интерфейса" уникальным образом определяет входной интерфейс, к которому относится входная ветвь (которому принадлежит дерево, подлежащее удалению). Поле 1224 "входные данные ключа" обеспечивает данные, которые сравниваются с входящим сообщением, чтобы определить, согласованы ли они с пересылающим узлом, подлежащим удалению, а поле 1226 "маска входного ключа" определяет соответствующие биты данных ключа при сравнении входных данных ключа с входящим пакетом. Резервное поле 1262 зарезервировано для использования в будущем, может устанавливаться на нуль передатчиком и игнорируется при этом приемником.

Если сообщение запроса "удалить дерево" протокола IFMP-C имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаги", и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает ответное сообщение "удалить дерево" протокола IFMP-C, которое представляет собой поля заголовка протокола IFMP-C (как показано на фиг.15b, но без поля "тело сообщения"), с флагом ACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение, указывающее отсутствие ошибки. Если сообщение запроса "удалить дерево" неуспешно, то IFMP-C исполнительное устройство возвращает соответствующее сообщение ошибки, которое представляет собой поля заголовка протокола IFMP-C (как показано на фиг.15b, но без поля "тело сообщения"), с флагом NACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение, указывающее причину сбоя (в качестве которых могут быть следующие: недействительный идентификатор

интерфейса, не существует ветвь с входным ключом, как определено в запросе; другая ветвь существует с тем же входным ключом, как определено, но с другими флагами, и т.п.).

На фиг. 18c показана структура сообщения запроса 1300 "переместить переход" протокола IFMP-C, содержащее многие из полей, описанных со ссылками на фиг.18a. Формат сообщения 1300 запроса "переместить переход" представляет собой обобщенный формат, ранее описанный со ссылками на фиг.15b, причем поле 1030 "тело сообщения" включает следующие поля: поле 1201 "идентификатор входного интерфейса", поле 1202 "входное предшествование", поле 1204 "входные флаги", 32-битовое поле 1303 "старый идентификатор выходного интерфейса", 24-битовое резервное поле 1304, 8-битовое поле 1210 "длина ключа", 8-битовое поле 1398 "длина старого выходного заголовка", 8-битовое поле 1310 "старая длина удаления", 8-битовое поле 1312 "старый тип преобразования", 8-битовое поле 1314 "длина старых данных преобразования", 32-битовое поле 1318 "старая обработка качества обслуживания", 32-битовое поле 1320 "идентификатор нового выходного интерфейса", 32-битовое резервное поле 1322, 8-битовое поле 1324 "длина нового выходного заголовка", 8-битовое поле 1326 "новая длина удаления", 8-битовое поле 1328 "новый тип преобразования", 8-битовое поле 1330 "длина новых данных преобразования", 32-битовое поле 1334 "новая обработка качества обслуживания", поле 1224 "входные данные ключа" предварительно определенной длины, поле 1226 "маска входного ключа" предварительно определенной длины, поле 1340 "данные старого выходного заголовка" предварительно определенной длины, поле 1342 "старые данные преобразования" предварительно определенной длины, поле 1344 "данные нового выходного заголовка" предварительно определенной длины, поле 1346 "новые данные преобразования" предварительно определенной длины. Многие из этих полей описаны выше со ссылками на фиг. 18a, а другие из полей сообщения запроса 1300 "переместить переход" легко понять, учитывая то, что значения полей, определяющие выходные данные старой ветви, заменяются значениями полей, определяющих выходные данные новой ветви, занимающей место старой ветви.

Если сообщение запроса "переместить переход" протокола IFMP-C имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаги", и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает ответное сообщение "переместить переход" протокола IFMP-C, которое имеет тот же формат 1250, что и ответное сообщение "добавить/удалить переход" протокола IFMP-C (как показано на фиг.18c), с флагом ACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение, указывающее отсутствие ошибки. Если сообщение запроса "переместить переход" безуспешно, то IFMP-C исполнительное устройство возвращает соответствующее сообщение ошибки, которое представляет собой поля

заголовка протокола IFMP-C (как показано на фиг. 15b, но без поля "тело сообщения"), с флагом NACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение, указывающее причину сбоя (в качестве которых могут быть следующие: недействительный идентификатор интерфейса, длина входного ключа больше, чем максимальная поддерживаемая длина, недостаточные ресурсы в IFMP-C исполнительном устройстве, выходное преобразование не поддерживается или не распознано, не существует ветвь с входным ключом, как определено для исходной ветви в запросе; исходная ветвь не существует, и ни одна ветвь не согласуется с новой ветвью; другая ветвь существует с тем же входным ключом, как определено, но с другими флагами, параметры качества обслуживания недействительны или не поддерживаются, специфическая пользовательская ошибка препятствует завершению запроса и т.п.).

Еще одно сообщение протокола IFMP-C, сообщение "получить статистику дерева", используется IFMP-C контроллером для определения того, когда узлы больше не используются, так чтобы эти узлы можно было восстановить. IFMP-C исполнительное устройство поддерживает счетчик, ведущий отсчет постоянно, пока узел используется для пересылки пакета. Такой узел сохраняется для каждого дерева, так как каждая ветвь такого дерева будет использоваться одно и то же число раз. Как видно из фиг.19a, сообщение "получить статистику дерева" протокола IFMP-C обобщенный формат, ранее описанный со ссылками на фиг. 15b, причем поле 1030 "тело сообщения" включает перечень информации, включающей в себя в конкретном варианте осуществления данные дерева: поле 1402 "данные 1 дерева", поле 1402 "данные 2 дерева". Дополнительные поля "данные дерева" могут быть включены в перечень, содержащийся в поле 1030 "тело сообщения". На фиг.19b показана структура 1406 поля "данные дерева", в которой используются все поля данных дерева. Каждая структура поля "данные дерева" включает следующие поля: поле 1201 "идентификатор входного интерфейса", поле 1202 "входное предшествование", поле 1204 "входные флаги", 40-битовое резервное поле 1408, поле 1210 "длина ключа", 18-битовое поле 1410 "размер записи", 64-битовое поле 1412 "отсчет использования", поле 1224 "входные данные ключа" предварительно определенной длины и поле 1226 "маска входного ключа" предварительно определенной длины. Поле 1201 "идентификатор входного интерфейса" уникальным образом определяет входной интерфейс, к которому относится входная ветвь (для которой должна быть получена статистика дерева). Резервное поле 1408 зарезервировано для использования в будущем, может быть установлено в нуль передатчиком и при этом будет игнорироваться приемником. Поле 1410 "размер записи", которое показывает размер конкретной записи дерева (например, в поле 1402 "данные 1 дерева"), используются для отыскания начала следующей записи дерева (например, в поле 1404 "данные 2 дерева").

Поле 1412 "отсчет использования" представляет собой 64-битовое целое число без знака, которое получает приращение всякий раз, когда IFMP-C исполнительное устройство использует конкретное дерево для пересылки пакета. В запросном сообщении поле 1412 "отсчет использования" устанавливается в нуль передатчиком и при этом будет игнорироваться приемником. Остальные поля не описаны здесь, поскольку они были описаны выше со ссылками на фиг.18a.

Если сообщение запроса 1400 "получить статистику дерева" протокола IFMP-C имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаги", и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает ответное сообщение "удалить дерево" протокола IFMP-C, которое имеет тот же самый формат, что и сообщение запроса 1400 (как показано на фиг.19a), с флагом ACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение, указывающее отсутствие ошибки. Ответное сообщение "получить статистику дерева" также возвращает в ответ значения соответствующих счетчиков в полях "отсчет использования" полей "данные дерева".

Если сообщение запроса "получить статистику дерева" безуспешно, то IFMP-C исполнительное устройство возвращает соответствующее сообщение ошибки, которое представляет собой поля заголовка протокола IFMP-C (как показано на фиг. 15b, но без поля "тело сообщения") сообщения запроса, с флагом NACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение, указывающее причину сбоя (в качестве которых могут быть следующие: недействительный идентификатор интерфейса, одна из определенных выходных ветвей не существует в IFMP-C исполнительном устройстве; другая ветвь существует с тем же входным ключом, как определено, но с другими флагами, и т.п.).

Еще одним сообщением протокола IFMP-C является сообщение "считать переход", которое используется для целей диагностики и отладки, для обеспечения возможности IFMP-C контроллеру получить данные для всех ветвей пересылки в IFMP-C исполнительном устройстве. Для нумерации всех ветвей в IFMP-C исполнительном устройстве сообщение "считать переход" использует операцию "получить следующее". Всякий раз, когда соответствующий элемент готов, IFMP-C исполнительное устройство возвращает информацию ветви, а также соответствующий идентификатор, используемый в качестве аргумента для следующей операции считывания. Этот идентификатор является непрозрачным для IFMP-C контроллера и используется IFMP-C исполнительным устройством для напоминания, где оно остановилось при последнем считывании. Предварительно определенное значение, например 0 в конкретном варианте осуществления, может быть зарезервировано и использовано IFMP-C контроллером для получения исходного элемента ввода в IFMP-C исполнительное устройство. Последующая

строка запросов получения узлов пересылки будет давать успешный результат до тех пор, пока все таблицы не будут перечислены, после чего IFMP-C исполнительное устройство выдаст указание того, что достигнут конец перечня.

На фиг. 20а и 20b показаны структуры сообщения 1420 запроса считывания ветви и сообщения 1430 ответа считывания ветви протокола IFMP-C соответственно. Как показано на фиг. 20а, сообщение 1420 запроса считывания ветви протокола IFMP-C имеет обобщенный формат, как описано выше со ссылками на фиг.15b, причем поле 1030 "тело сообщения" содержит поле 1201 "идентификатор входного интерфейса" и 32-битовое поле 1422 "следующий идентификатор". Поле 1201 "идентификатор входного интерфейса" уникальным образом определяет входной интерфейс, на котором должны считываться ветви. Поле 1422 "следующий идентификатор" является непрозрачным 32-битовым значением, возвращенным в виде части предыдущего ответного сообщения "считать ветвь". Значение в поле 1422 "следующий идентификатор" представляет состояние, поддерживаемое IFMP-C исполнительным устройством, для поддержания отслеживания позиции последнего запроса "считать переход". Предварительно определенное значение, например, 0 в конкретном варианте осуществления, используется для указания того, что запуск должен быть осуществлен с начала списка.

Если сообщение запроса "считать переход" протокола IFMP-C имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаги", и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает ответное сообщение "считать переход" протокола IFMP-C (фиг.20b) с флагом ACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение, указывающее отсутствие ошибки. Как показано на фиг. 20b, ответное сообщение "считать переход" протокола IFMP-C имеет обобщенный формат, ранее рассмотренный со ссылками на фиг.15b, причем поле "тело сообщения" 1030 содержит следующие поля: поле 1201 "идентификатор входного интерфейса", поле 1202 "входное вхождение", поле 1204 "входные флаги", поле 1206 "идентификатор выходного интерфейса", 24-битовое резервное поле 1432, поле 1210 "длина ключа", поле 1212 "длина выходного заголовка", поле 1214 "длина удаления", поле 1216 "тип преобразования", поле 1218 "длина данных преобразования", поле 1222 "обработка качества обслуживания", поле 1422 "следующий идентификатор", поле 1224 "входные данные ключа" предварительно определенной длины, поле 1226 "маска входного ключа" предварительно определенной длины, поле 1228 "данные выходного заголовка" предварительно определенной длины, поле 1230 "данные преобразования" предварительно определенной длины. Резервное поле 1432 зарезервировано для использования в будущем, устанавливается на нуль передатчиком и игнорируется приемником. Поле 1422 "следующий идентификатор"

представляет состояние, поддерживаемое IFMP-C исполнительным устройством, для отслеживания позиции последнего запроса, и используется в качестве ввода в следующее сообщение "считать переход". Остальные поля были описаны выше со ссылками на фиг.18а. Если запрос "считать переход" был неуспешным, то IFMP-C исполнительное устройство возвращает соответствующее сообщение ошибки, которое состоит из заголовка сообщения запроса "считать переход" протокола IFMP-C, за исключением того, что флаг NACK установлен в поле 1020 "флаги", и в поле 1018 "код" установлено предварительно определенное значение, указывающее причину сбоя (в качестве которых могут быть следующие: недействительный идентификатор интерфейса в сообщении, недействительный следующий идентификатор, IFMP-C исполнительное устройство не обеспечивает выделения буфера сообщения для завершения ответа, специфическая пользовательская ошибка препятствует завершению запроса и т.п.).

Помимо сообщений протоколов IFMP-C, определяющих близость, интерфейсы и ветви, сообщения протокола IFMP-C управляющие сообщения, такие как сообщение сброса. Сообщение сброса протокола IFMP-C используется для повторной инициализации состояния IFMP-C исполнительного устройства, когда IFMP-C контроллер потерял связь с IFMP-C исполнительным устройством или когда IFMP-C контроллер предполагает, что в IFMP-C исполнительном устройстве произошел сбой. Сообщение сброса протокола IFMP-C выдает команду IFMP-C исполнительному устройству сбросить все его состояния с установкой в исходное состояние. После приема сообщения сброса IFMP-C исполнительное устройство удаляет все ветви пересылки и инициализирует интерфейсы без сброса состояния протокола близости IFMP-C. Сообщение сброса состоит из заголовка IFMP-C, показанного на фиг.15b (без поля "тело сообщения"), причем поле 1016 "тип сообщения" установлено для обозначения данного сообщения в качестве сообщения сброса. Если сообщение запроса "сброс" протокола IFMP-C имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаги", и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает ответное сообщение "сброс" протокола IFMP-C, которое состоит из идентичного заголовка IFMP-C, что и в сообщении запроса "сброс", но с флагом ACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение, указывающее отсутствие ошибки. Если сообщение "сброс" неуспешно, то IFMP-C исполнительное устройство возвращает ответное сообщение "сброс" протокола IFMP-C, которое состоит из идентичного заголовка IFMP-C, что и в сообщении запроса "сброс", но с флагом NACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение, указывающее причину сбоя (например, специфическая пользовательская ошибка, препятствующая завершению запроса).

Помимо сообщений протоколов IFMP-C, определяющих близость, интерфейсы и ветви, сообщения протокола IFMP-C также включают управляющие сообщения, которые используются для получения информации, требуемой для управления сетью и целей диагностики. Управляющие сообщения протокола IFMP-C включают в себя различные сообщения: сообщения информации узла и сообщения статистики интерфейса.

Сообщения информации узла протокола IFMP-C обеспечивают получение информации (например, номер версии программного обеспечения и т.п.) об узле, выполняющем протокол IFMP-C. Сообщение запроса информации узла имеет обобщенный формат заголовка формата IFMP-C (без поля 1030 "тело сообщения", как показано на фиг.15b), с полем 1016 "тип сообщения", идентифицирующим сообщение, как сообщение информации узла.

Если сообщение запроса "информация узла" протокола IFMP-C имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаги", и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает ответное сообщение 1440 "информация узла" протокола IFMP-C (фиг.21a) с флагом ACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение (например, 0 в конкретном варианте осуществления), указывающее отсутствие ошибки. Как показано на фиг.21a, ответное сообщение 1440 "информация узла" протокола IFMP-C имеет обобщенный формат, ранее рассмотренный со ссылками на фиг.15b, причем поле "тело сообщения" 1030 содержит следующие поля: 48-битовое поле 1442 "идентификатор узла", 48-битовое поле 1444 "идентификатор порождающего элемента", 16-битовое поле 1446 "тип узла", 15-битовое резервное поле 1448, 32-битовое поле 1450 "порождающий слот", 32-битовое поле 1452 "порождающая стойка", 16-битовое поле 1454 "минимальная версия программного обеспечения", 16-битовое поле 1456 "максимальная версия программного обеспечения". Поле 1442 "идентификатор узла" представляет собой значение (например, MAC адрес узла), который однозначно определяет узел. Для ситуаций, когда узел является частью большей системы, например платы на шасси, поле "идентификатор порождающего элемента" 1444 устанавливается на значение (например, MAC адрес для порождающего элемента), который однозначно идентифицирует контейнер (или порождающий элемент) узла. Если узел не является частью большей системы, такой как плата на шасси, поле 1444 "идентификатор порождающего элемента" устанавливается на предварительно определенное значение, например 0 в конкретном варианте осуществления, указывающее отсутствие порождающего узла. Поле 1446 "тип узла" представляет собой 16-битовое целое число без знака, которое указывает тип IFMP-C узла. Значение поля 1446 "тип узла" может быть присвоено поставщиком, так что комбинация части "уникального идентификатора организации" (OUI, который представляет собой 24 старших бита MAC-адреса) поля "идентификатор узла"

и поля "тип узла" является уникальной для каждого типа IFMP-C исполнительного устройства. Резервное поле 1448 зарезервировано для использования в будущем, может быть установлено в 0 передатчиком и игнорироваться приемником. Если узел является частью большей системы, такой как плата на шасси, то поле 1450 "слот порождающего элемента" устанавливается в соответствии со слотом, который узел занимает в порождающем контейнере. Если IFMP-C исполнительное устройство не может определить информацию слота или если не является частью более крупного контейнера, то поле 1450 "слот порождающего элемента" может быть установлен в нуль. Если узел является частью более крупной системы, например платы в шасси, то поле 1452 "стойка порождающего элемента" устанавливается в соответствии со слотом, который узел занимает в этом порождающем контейнере. Если IFMP-C исполнительное устройство не может определить информацию слота, или если не является частью более крупного контейнера, то поле 1452 "стойка порождающего элемента" может быть установлен в нуль. Если поле 1454 "минимальная версия программного обеспечения" указывает минимальное видоизменение программного обеспечения, которое выполняется на IFMP-C исполнительном устройстве, а поле 1456 "максимальная версия программного обеспечения" указывает максимальное видоизменение программного обеспечения, которое выполняется на IFMP-C исполнительном устройстве.

Если запрос "информация узла" неуспешен, то IFMP-C исполнительное устройство возвращает сообщение ошибки "информации узла" протокола IFMP-C, которое состоит из идентичного заголовка IFMP-C. Сообщение ошибки "информация узла" должно быть идентичным заголовку IFMP-C в сообщении запроса "информация узла", но с флагом NACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение, указывающее причину сбоя. Причинами сбоя могут быть невозможность выделения IFMP-C исполнительным устройством буфера сообщений для завершения ответа, специфическая пользовательская ошибка, препятствующая завершению запроса или иные причины.

К другим сообщениям сетевого управления протокола IFMP-C относятся сообщения "статистика интерфейсов" протокола IFMP-C, используемые для получения информации об интерфейсах IFMP-C исполнительного устройства. Сообщение 1460 запроса статистики интерфейсов протокола IFMP-C позволяет IFMP-C контроллеру запрашивать статистику для более чем одного интерфейса в одном сообщении запроса. Как показано на фиг.21b, сообщение 1460 запроса статистики интерфейсов протокола IFMP-C имеет обобщенный формат, показанный на фиг.15b, причем поле 1030 "тело сообщения" включает следующие поля: 16-битовое резервное поле 1462, 16-битовое поле 1464 "число интерфейсов", за которым следуют поля 1466, 1468 и т.д. множества идентификаторов

интерфейсов. Резервное поле 1462 зарезервировано для использования в будущем, оно может быть установлено на 0 передатчиком и при этом будет игнорироваться приемником. Поле 1464 "число интерфейсов" указывает число полей "идентификатор интерфейса" в сообщении запроса. Сообщение запроса 1460 "статистика интерфейсов" включает множество полей "идентификатор интерфейса" (например 1466, 1468) в идентификаторы списка для каждого интерфейса, который представляет интерес для IFMP-C контроллера. В конкретном варианте осуществления ответ может быть выдан в одном сообщении, так что IFMP-C исполнительное устройство возвращает ответы для того числа интерфейсов, которым соответствует одно ответное сообщение.

Если сообщение запроса "статистика интерфейсов" протокола IFMP-C имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаги", и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает ответное сообщение 1470 "статистика интерфейсов" протокола IFMP-C (фиг.21c) с флагом ACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение (например, 0 в конкретном варианте осуществления), указывающее отсутствие ошибки. Как показано на фиг. 21c, ответное сообщение 1470 "статистика интерфейсов" протокола IFMP-C имеет обобщенный формат, ранее рассмотренный со ссылками на фиг.15b, причем поле "тело сообщения" 1030 содержит поля "статистика интерфейсов" (например, 1472, 1473), которые обеспечивают общую статистику (общую для различных типов интерфейсов) и специальную статистику (которая применима только к специфическим типам интерфейсов). Типовая структура 1480 поля "статистика интерфейсов", как показано на фиг.21d, включает следующие поля: 8-битовое резервное поле 1482, 8-битовое поле 1484 "тип интерфейса", 16-битовое поле 1486 "длина записи", 32-битовое поле 1488 "идентификатор интерфейса", 16-битовое поле 1490 "длина общей статистики", поле 1492 "длина специальной статистики", поле 1494 "общая статистика", поле 1496 "специальная статистика". Резервное поле 1482 зарезервировано для использования в будущем, оно может быть установлено на 0 передатчиком и будет игнорироваться приемником. Поле 1484 "тип интерфейса" описывает тип интерфейса (например, интерфейс ATM, Ethernet или другой локальной сети), о котором сделан запрос. Поле 1488 "идентификатор интерфейса" однозначно указывает, какой интерфейс описывают статистические данные. Поле 1490 "длина общей статистики" определяет длину поля 1494 "общая статистика" в байтах, а поле 1492 "длина специальной статистики" определяет длину поля 1496 "специальная статистика" в байтах. Поле 1494 "общая статистика" содержит общие статистические данные, связанные со всеми интерфейсами, и имеет структуру, показанную на фиг.21e. Поле 1496 "специальная статистика" содержит статистические данные для специальных типов интерфейсов и имеет структуру,

показанную на примерах интерфейсов ATM и Ethernet на фиг.21f и 2 g. Для интерфейсов других локальных сетей могут использоваться другие структуры.

Как показано на фиг.21e, поле 1494 "общая статистика" содержит 64-битовое поле 1500 "принятые байты", которое указывает число байтов, принятых на определенном интерфейсе; 64-битовое поле 1502 "принятые пакеты групповой передачи", которое указывает число пакетов, адресованных как пакеты групповой передачи, которые были приняты на конкретном интерфейсе; 64-битовое поле 1504 "принятые пакеты широковещательной передачи", которое указывает число пакетов, адресованных как пакеты широковещательной передачи, которые были приняты на конкретном интерфейсе; 64-битовое поле 1506 "принятые пакеты одноадресной передачи", которое указывает число пакетов, переданных как однократные пакеты, принятые на конкретном интерфейсе; 64-битовое поле "принятые отказы", которое указывает число пакетов, которые были отброшены на входе данного интерфейса; 64-битовое поле 1510 "принятые ошибки", которое указывает число принятых ошибок на конкретном интерфейсе; 64-битовое поле 1512 "принятые неопознанные элементы", которое указывает число принятых пакетов, имевших неопознанный протокол, на конкретном интерфейсе; 64-битовое поле 1514 "переданные байты", которое указывает число байтов, переданных на определенном интерфейсе; 64-битовое поле 1516 "переданные пакеты групповой передачи", которое указывает число пакетов, адресованных как пакеты групповой передачи, которые были переданы на конкретном интерфейсе; 64-битовое поле 1518 "переданные пакеты широковещательной передачи", которое указывает число пакетов, адресованных как пакеты широковещательной передачи, которые были переданы на конкретном интерфейсе; 64-битовое поле 1520 "переданные пакеты одноадресной передачи", которое указывает число пакетов, переданных как однократные пакеты, переданные на конкретном интерфейсе; 64-битовое поле 1522 "переданные отказы", которое указывает число пакетов, которые были отброшены при передаче на входе данного интерфейса; 64-битовое поле 1524 "переданные ошибки", которое указывает число ошибок при передаче на конкретном интерфейсе.

Поле 1496 "специальная статистика" для определенных статистических данных для ATM интерфейса и для Ethernet интерфейса показано на фиг.21f и 21g. Как показано на фиг.21f, структура поля 1530 "специальная статистика" для ATM интерфейса содержит 64-битовое поле 1532 "принятые элементы данных", которое указывает число ATM элементов данных, принятых на конкретном интерфейсе; 64-битовое поле 1534 "переданные элементы данных", которое указывает число ATM элементов данных, переданных на конкретном интерфейсе; 64-битовое поле "AAL5 ошибки ЦИК", которое указывает число пакетов с некорректным значением контрольной суммы при проверке циклическим избыточным кодом на уровне 5

АТМ адаптации; и 64-битовое поле 1536 "физические ошибки", которое указывает число физических ошибок, которые имели место на определенном АТМ интерфейсе. Другой пример поля 1496 "специальная статистика" для интерфейса Ethernet включает: 64-битовое поле 1542 "принятые ошибки ЦИК", которое указывает число пакетов, которые были приняты с ненадлежащей контрольной суммой при проверке с использованием циклического избыточного кода на конкретном интерфейсе; и 64-битовое поле 1544 "конфликты при передаче", которое указывает число конфликтов, имевших место при попытках передать пакеты на конкретном интерфейсе.

Если запрос "статистика интерфейсов" неуспешен, то IFMP-C исполнительное устройство возвращает сообщение ошибки "статистика интерфейсов" протокола IFMP-C, которое состоит из идентичного заголовка IFMP-C. Сообщение ошибки "статистика интерфейсов" должно быть идентичным заголовку IFMP-C в сообщении запроса "информация узла", но с флагом NACK, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предварительно определенное значение, указывающее причину сбоя. Причинами сбоя могут быть следующие: недействительный идентификатор интерфейса, приведенное в перечне число интерфейсов не соответствует числу интерфейсов в сообщении, невозможность выделения IFMP-C исполнительным устройством буфера сообщений для завершения ответа, специфическая пользовательская ошибка, препятствующая завершению запроса или иные причины.

4. Заключение

Заявленное изобретение направлено на создание усовершенствованного способа и устройства для передачи пакетов в сети. Следует иметь в виду, что вышеприведенное описание является иллюстративным и не накладывает никаких ограничений. Различные варианты осуществления должны быть очевидными для специалистов в данной области техники, как вытекающие из сведений, приведенных в описании. Для примера изобретение проиллюстрировано в применении к передаче IP-пакетов, обеспечивающих пересылку речевых данных, видеосигналов, данных изображений, факсимильных данных и сигналов данных, однако изобретение не ограничивается приведенными видами сигналов. Кроме того, изобретение проиллюстрировано в связи с конкретными компонентами и рабочими скоростями, однако изобретение не ограничивается конкретными указанными значениями параметров. Следует иметь в виду, что различные конкретные примеры типов сообщений, ошибок и т.п. приведены только для конкретных вариантов осуществления, являющихся возможными вариантами, при этом в изобретении могут использоваться и другие дополнительные или отличные признаки или комбинации таких признаков. Объем изобретения должен определяться не на основе информации, приведенной в описании, а только в соответствии с формулой изобретения, совместно с полным объемом эквивалентных средств, на которые распространяются

пункты формулы изобретения.

Формула изобретения:

1. Способ передачи пакетов данных между узлом восходящей линии связи и узлом нисходящей линии связи в сети, причем указанный узел нисходящей линии связи расположен в направлении нисходящего потока относительно узла восходящей линии связи, включающий этапы установления по умолчанию виртуальных каналов между упомянутым узлом восходящей линии связи и упомянутым узлом нисходящей линии связи, приема пакета данных в упомянутом узле нисходящей линии связи, определения того, поступил ли пакет данных по одному из упомянутых установленных по умолчанию виртуальных каналов, селективного выполнения классификации потока в узле нисходящей линии связи по упомянутому пакету данных, выбора в упомянутом узле нисходящей линии связи свободной метки для узла восходящей линии связи, информирования упомянутого узла восходящей линии связи о том, что последующие пакеты данных, принадлежащие к упомянутому определенному потоку данных, должны передаваться с указанной выбранной свободной меткой.

2. Способ по п. 1, отличающийся тем, что упомянутые узлы восходящей линии связи и нисходящей линии связи используют асинхронный режим передачи.

3. Способ по п. 2, отличающийся тем, что упомянутая свободная метка содержит идентификатор виртуального маршрута и идентификатор виртуального канала.

4. Способ по п. 1, отличающийся тем, что упомянутая сеть включает локальную компьютерную сеть.

5. Способ передачи пакетов данных между узлом восходящей линии связи и узлом нисходящей линии связи в сети, причем указанный узел нисходящей линии связи расположен в направлении нисходящего потока относительно узла восходящей линии связи, включающий этапы установления по умолчанию виртуальных каналов между упомянутым узлом восходящей линии связи и упомянутым узлом нисходящей линии связи, приема пакета данных в упомянутом узле нисходящей линии связи, выполнения классификации потока в упомянутом узле нисходящей линии связи по упомянутому пакету данных для определения того, принадлежит ли этот пакет данных конкретному потоку, который должен быть переадресован в узел восходящей линии связи, выбора в упомянутом узле нисходящей линии связи свободной метки, информирования упомянутого узла восходящей линии связи о том, что последующие пакеты данных, принадлежащие к упомянутому определенному потоку данных, должны передаваться с указанной выбранной свободной меткой, причем упомянутая сеть содержит локальную компьютерную сеть, а этап информирования выполняется с использованием программного обеспечения протокола управления потоком IFMP, который обеспечивает информационный обмен между упомянутыми узлами восходящей линии связи и нисходящей линии связи.

6. Способ передачи пакетов данных между

узлом восходящей линии связи и узлом нисходящей линии связи в сети, причем указанный узел нисходящей линии связи расположен в направлении нисходящего потока относительно узла восходящей линии связи, включающий этапы установления по умолчанию виртуальных каналов между упомянутым узлом восходящей линии связи и упомянутым узлом нисходящей линии связи, приема пакета данных в упомянутом узле нисходящей линии связи, выполнения классификации потока в упомянутом узле нисходящей линии связи по упомянутому пакету данных для определения того, принадлежит ли этот пакет данных конкретному потоку, который должен быть переадресован в узел восходящей линии связи, выбора в упомянутом узле нисходящей линии связи свободной метки, информирования упомянутого узла восходящей линии связи о том, что последующие пакеты данных, принадлежащие к упомянутому определенному потоку данных, должны передаваться с указанной выбранной свободной меткой, при этом этап выполнения классификации потока включает анализ идентификатора потока пакета данных для определения того, принадлежит ли указанный пакет данных к определенному потоку, который должен быть переадресован в узел восходящей линии связи, причем указанный идентификатор потока содержит определенные поля заголовка из этого пакета данных, и передачи упомянутых последующих пакетов данных, принадлежащих к определенному потоку, причем упомянутые определенные поля заголовка удалены для обеспечения защищенности.

7. Способ по п. 6, отличающийся тем, что упомянутые последующие пакеты могут быть восстановлены с использованием идентификатора потока в месте назначения, предназначенном для приема указанного определенного потока.

8. Способ коммутации потока в первом узле, причем указанный первый узел имеет нисходящую линию связи к второму узлу и восходящую линию связи к третьему узлу, включающий этапы выполнения классификации потока в первом узле по первому пакету данных для определения того, принадлежит ли этот первый пакет данных к определенному потоку, который должен быть переадресован в третьем узле, выбора первой свободной метки в упомянутом первом узле, информирования упомянутого третьего узла о том, что последующие пакеты, принадлежащие к упомянутому определенному потоку, должны передаваться с указанной выбранной первой свободной меткой, выполнения классификации потока во втором узле по второму пакету данных для определения того, принадлежит ли этот второй пакет данных к определенному потоку, который должен быть переадресован в третьем узле, выбора второй свободной метки в упомянутом втором узле, информирования упомянутого первого узла о том, что последующие пакеты данных, принадлежащие к упомянутому определенному потоку, должны передаваться с указанной выбранной второй свободной меткой, селективной коммутации упомянутого определенного потока посредством

упомянутого первого узла из восходящей линии связи в нисходящую линию связи, при этом упомянутый определенный поток из восходящей линии связи может передаваться посредством первого узла в нисходящую линию связи.

9. Способ по п. 8, отличающийся тем, что в качестве второго пакета данных используют первый пакет данных.

10. Способ по п. 9, отличающийся тем, что упомянутые первый, второй и третий узлы используют асинхронный режим передачи.

11. Способ по п. 10, отличающийся тем, что упомянутая первая и вторая свободные метки обе содержат идентификатор виртуального маршрута и идентификатор виртуального канала.

12. Способ коммутации потока в первом узле, причем указанный первый узел имеет нисходящую линию связи к второму узлу и восходящую линию связи к третьему узлу, включающий этапы выполнения классификации потока в первом узле по первому пакету данных для определения того, принадлежит ли этот первый пакет данных определенному потоку, который должен быть переадресован в третьем узле, выбора первой свободной метки в упомянутом первом узле, информирования упомянутого третьего узла о том, что последующие пакеты данных, принадлежащие упомянутому определенному потоку, должны передаваться с указанной выбранной первой свободной меткой, выполнения классификации потока во втором узле по второму пакету данных для определения того, принадлежит ли этот второй пакет данных определенному потоку, который должен быть переадресован в третьем узле, выбора второй свободной метки в упомянутом втором узле, информирования упомянутого первого узла о том, что последующие пакеты данных, принадлежащие упомянутому определенному потоку, должны передаваться с указанной выбранной второй свободной меткой, при этом упомянутый определенный поток из упомянутой восходящей линии связи может коммутироваться посредством упомянутого первого узла в нисходящую линию связи, упомянутый второй пакет данных представляет собой упомянутый первый пакет данных, а этапы информирования выполняются с использованием программного обеспечения протокола управления потоком IFMP, который обеспечивает информационный обмен между упомянутыми узлами.

13. Способ передачи пакетов данных между узлом восходящей линии связи и узлом нисходящей линии связи в сети, включающий этапы приема пакета данных от узла восходящей линии связи в узле нисходящей линии связи, выполнения классификации потока в упомянутом узле нисходящей линии связи по упомянутому пакету данных, селективного присвоения потоку данных в упомянутом узле нисходящей линии связи свободной метки, выбранной из свободных меток узла восходящей линии связи, передачи к узлу восходящей линии связи сообщения, указывающего упомянутую свободную метку, и селективного включения упомянутой свободной метки в последующие пакеты данных от упомянутого узла

восходящей линии связи.

14. Способ по п. 13, отличающийся тем, что упомянутая свободная метка содержит идентификатор виртуального маршрута и идентификатор виртуального канала.

15. Способ по п. 13, отличающийся тем, что упомянутые узлы восходящей линии связи и нисходящей линии связи используют асинхронный режим передачи.

16. Способ передачи пакетов данных от первого узла ко второму узлу в сети, включающий этапы приема первого пакета данных от упомянутого первого узла, выполнения классификации потока по упомянутому первому пакету данных, селективного присвоения свободной метки потоку, связанному с первым пакетом данных, передачи к первому узлу сообщения, указывающего упомянутую свободную метку, и направления последующих пакетов данных из упомянутого потока с использованием канала, указанного упомянутой свободной меткой.

17. Способ по п. 16, отличающийся тем, что дополнительно включает этапы установки времени таймера после присвоения упомянутой свободной метки и направления последующих пакетов данных из упомянутого потока с использованием установленного по умолчанию канала после истечения установленного времени таймера.

18. Способ коммутации потока в первом узле, включающий этапы приема первого пакета данных упомянутого потока от узла восходящей линии связи, выполнения классификации потока в первом узле по первому пакету данных, выбора первой свободной метки для упомянутого узла восходящей линии связи, передачи упомянутому узлу восходящей линии связи сообщения, указывающего упомянутую первую свободную метку, приема от узла нисходящей линии связи сообщения, указывающего вторую свободную метку, и селективной коммутации последующих пакетов упомянутого потока от упомянутого узла восходящей линии связи к упомянутому узлу нисходящей линии связи на основе упомянутых первой и второй свободных меток.

19. Способ по п. 18, отличающийся тем, что упомянутые первый, второй и третий узлы используют асинхронный режим передачи.

20. Базовый коммутационный блок в системе для передачи пакетов данных в сети, содержащий аппаратное средство коммутации, контроллер, связанный с упомянутым аппаратным средством коммутации, причем контроллер содержит процессор и память и предназначен для управления упомянутым аппаратным средством коммутации, программное обеспечение, хранящееся на материальном носителе, обеспечивающее возможность базовому коммутационному блоку осуществлять динамический сдвиг между маршрутизацией и коммутацией пакетов для оптимизации пропускной способности пакетного трафика.

21. Базовый коммутационный блок по п. 20, отличающийся тем, что упомянутое программное обеспечение использует классификацию потока.

22. Базовый коммутационный блок по п. 21, отличающийся тем, что упомянутое

аппаратное средство коммутации использует коммутацию на основе асинхронного режима передачи.

23. Базовый коммутационный блок по п. 22, отличающийся тем, что упомянутая классификация потока использует в качестве меток идентификатор виртуального маршрута и идентификатор виртуального канала.

24. Базовый коммутационный блок по п. 22, отличающийся тем, что упомянутое программное обеспечение включает первую подсистему программного обеспечения, установленную на контроллере, для осуществления информационного обмена и управления упомянутым аппаратным средством коммутации.

25. Базовый коммутационный блок по п. 24, отличающийся тем, что упомянутое программное обеспечение включает вторую подсистему программного обеспечения, обеспечивающую информационный обмен между двумя упомянутыми базовыми коммутационными блоками и определение формата для сообщений переадресации потока и подтверждений приема.

26. Базовый коммутационный блок по п. 25, отличающийся тем, что базовый коммутационный блок осуществляет локальное принятие решений классификации и решений в ответ на сообщение переадресации.

27. Базовый коммутационный блок по п. 20, отличающийся тем, что упомянутая сеть включает локальную сеть, содержащую компьютеры.

28. Базовый коммутационный блок по п. 24, отличающийся тем, что первая подсистема программного обеспечения включает протокол управления потоком IFMP.

29. Базовый коммутационный блок по п. 28, отличающийся тем, что вторая подсистема программного обеспечения включает протокол, подсистема программного обеспечения включает общий протокол управления коммутацией GSNP.

30. Базовый коммутационный блок по п. 22, отличающийся тем, что упомянутое программное обеспечение включает средство обработки качества обслуживания.

31. Базовый коммутационный блок по п. 21, отличающийся тем, что упомянутое аппаратное средство коммутации использует быстрореагирующую пакетную технологию.

32. Базовый коммутационный блок по п. 21, отличающийся тем, что упомянутое аппаратное средство коммутации использует технологию ретрансляции кадра.

33. Базовый коммутационный блок по п. 21, отличающийся тем, что упомянутое средство коммутации использует технологию Gigabit Ethernet.

34. Шлюзовой блок коммутации в системе для передачи пакетов в сети, причем упомянутая система включает базовый коммутационный блок, связанный с упомянутым шлюзовым коммутационным узлом посредством линии связи, содержащий контроллер шлюзового блока, содержащий процессор, память и множество плат сетевых интерфейсов, программное обеспечение, хранящееся на материальном носителе, обеспечивающее возможность шлюзовому блоку коммутации переадресовывать поток пакетов данных к упомянутому базовому коммутационному блоку, чтобы осуществлять

динамический сдвиг между маршрутизацией и коммутацией пакетов для оптимизации пропускной способности пакетного трафика.

35. Шлюзовой блок коммутации по п. 34, отличающийся тем, что упомянутое программное обеспечение использует классификацию потока.

36. Шлюзовой блок коммутации по п. 34, отличающийся тем, что упомянутый базовый коммутационный блок использует коммутацию на основе асинхронного режима передачи.

37. Шлюзовой блок коммутации по п. 36, отличающийся тем, что упомянутый шлюзовой блок коммутации и упомянутый базовый коммутационный блок используют в качестве меток идентификатор виртуального маршрута и идентификатор виртуального канала.

38. Шлюзовой блок коммутации по п. 37, отличающийся тем, что упомянутое программное обеспечение включает первую подсистему программного обеспечения, установленную на контроллере шлюзового блока, для осуществления информационного обмена между упомянутым шлюзовым блоком коммутации и упомянутым базовым коммутационным блоком в упомянутой системе и определения формата для сообщений переадресации потока и подтверждений приема.

39. Шлюзовой блок коммутации по п. 38, отличающийся тем, что упомянутый шлюзовой блок коммутации осуществляет локальное принятие решений о классификации потока и решений в ответ на сообщение переадресации.

40. Шлюзовой блок коммутации по п. 39, отличающийся тем, что первая подсистема программного обеспечения включает протокол управления потоком IFMP.

41. Шлюзовой блок коммутации по п. 39, отличающийся тем, что упомянутый базовый коммутационный блок использует технологию Gigabit Ethernet.

42. Исполнительное устройство коммутации в системе для передачи пакетов данных в сети, причем упомянутая система включает базовый коммутационный блок, связанный с упомянутым исполнительным устройством коммутации посредством линии связи, а базовый коммутационный блок содержит контроллер и процессор коммутации, содержащее процессор, память и множество плат сетевых интерфейсов, причем конкретная одна из указанного множества плат сетевых интерфейсов обеспечивает упомянутую линию связи, а по меньшей мере одна из упомянутого множества плат сетевых интерфейсов имеет возможность связи по меньшей мере с одним узлом сети, и компьютерно-читаемый программный код, хранящийся на материальном компьютерно-читаемом носителе, содержащем упомянутую память, при этом упомянутый компьютерно-читаемый программный код обеспечивает возможность контроллеру базового коммутационного блока классифицировать поток и переадресовывать указанный поток пакетов данных из первого узла во второй узел сети, и упомянутый компьютерно-читаемый программный код обеспечивает возможность контроллеру базового коммутационного блока выдавать команды упомянутому исполнительному

устройству коммутации для выполнения пересылки упомянутого потока от первого узла ко второму узлу через упомянутый процессор коммутации, при этом разгружая пересылку пакетов данных от контроллера к базовому коммутационному блоку.

43. Исполнительное устройство коммутации по п. 42, отличающееся тем, что первый узел соединен через первую одну из множества плат сетевых интерфейсов с упомянутым исполнительным устройством коммутации, а второй узел выбран из группы, включающей в себя другое исполнительное устройство коммутации, другой базовый коммутационный блок, шлюзовой блок коммутации или главный компьютер, при этом второй узел связан с процессором коммутации упомянутого базового коммутационного блока.

44. Исполнительное устройство коммутации по п. 42, отличающееся тем, что первый узел выбран из группы, включающей в себя другое исполнительное устройство коммутации, другой базовый коммутационный блок, шлюзовой блок коммутации или главный компьютер, при этом второй узел связан через первую одну из множества плат сетевых интерфейсов с упомянутым исполнительным устройством коммутации, а упомянутый компьютерно-читаемый программный код обеспечивает возможность упомянутому контроллеру базового коммутационного блока выдавать команду исполнительному устройству коммутации о том, как обрабатывать упомянутые пакеты данных в потоке, принятом от упомянутого процессора коммутации.

45. Исполнительное устройство коммутации по п. 42, отличающееся тем, что упомянутый процессор коммутации использует режим асинхронной передачи, ретрансляцию кадров, быстродействующую пакетную коммутацию, Ethernet со скоростью 10 Мб/с, Ethernet со скоростью 100 Мб/с или Gigabit Ethernet.

46. Исполнительное устройство коммутации по п. 42, отличающееся тем, что по меньшей мере одна из упомянутого множества плат сетевых интерфейсов представляет собой плату сетевого интерфейса Ethernet.

47. Исполнительное устройство коммутации по п. 42, отличающееся тем, что по меньшей мере одна из упомянутого множества плат сетевых интерфейсов представляет собой плату сетевого интерфейса сети асинхронного режима передачи.

48. Исполнительное устройство коммутации по п. 42, отличающееся тем, что упомянутый компьютерно-читаемый программный код содержит первую подсистему, установленную на упомянутом контроллере, и вторую подсистему, установленную в памяти исполнительного устройства коммутации, при этом упомянутые первая и вторая подсистемы обеспечивают информационный обмен между упомянутыми исполнительным устройством коммутации и базовым коммутационным блоком в упомянутой системе.

49. Исполнительное устройство коммутации по п. 48, отличающееся тем, что упомянутое исполнительное устройство

коммутации служит в качестве подчиненного узла по отношению к базовому коммутационному блоку, который осуществляет локальное принятие решений о классификации потока и решений в ответ на сообщение переадресации.

50. Исполнительное устройство коммутации по п. 48, отличающееся тем, что

упомянутый компьютерно-читаемый программный код содержит программное обеспечение протокола управления потоком для клиентов IFMP-C.

Приоритет по пунктам:

31.01.1996 по пп. 1-23, 27-50;

22.11.1996 по пп. 24-26.

5

10

15

20

25

30

35

40

45

50

55

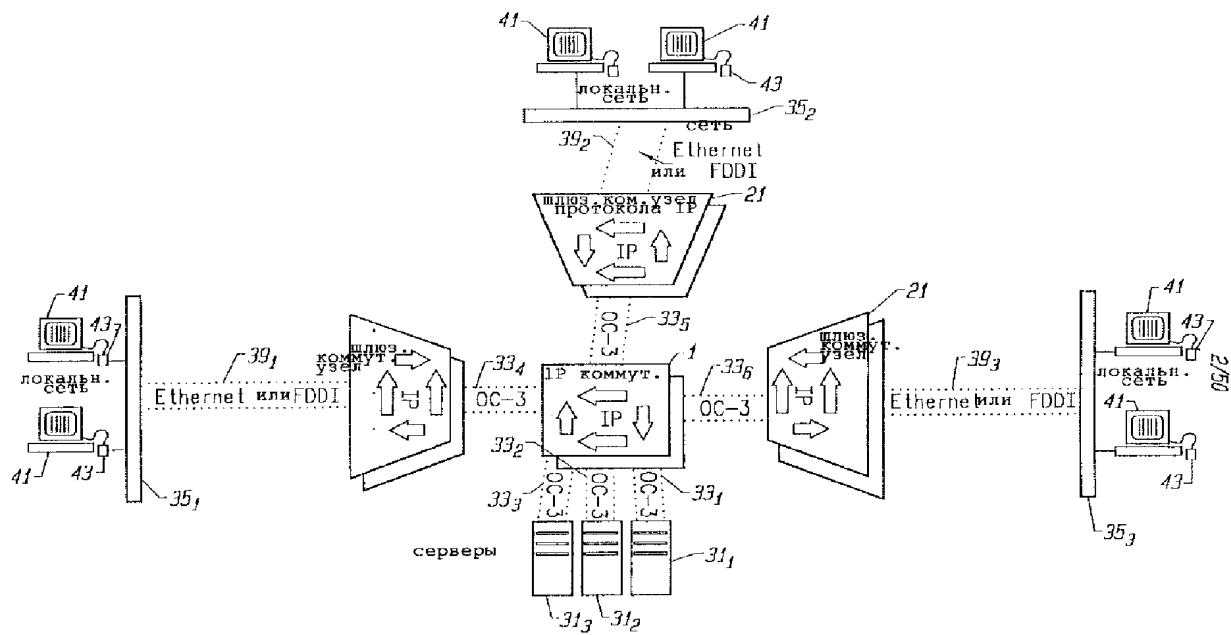
60

Микропроцессор	Процессор Intel Pentium, 133 МГц
Системная память	ЗУПВ 16 Мбайт/Кэш 256 К
Материнская плата	Материнская плата Intel Endeavor
Плата АТМ интерфейса	Zeitnet PCI АТМ NIC (155 Мб/с)
Стационарный или жесткий диск	Диск IDE 500 Мбайт
Накопители	Для стандартных гибких и CD-ROM
Источник питания	Стандартный источник питания
Шасси	Стандартное Шасси

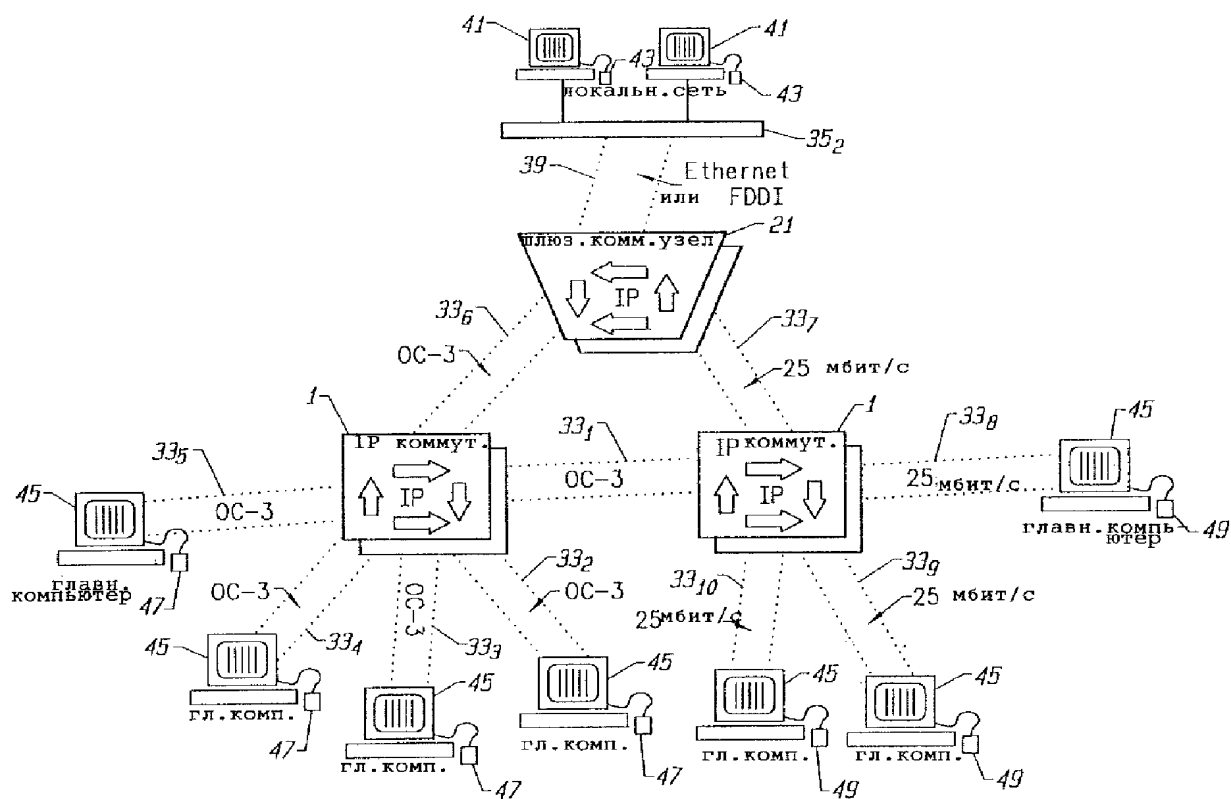
RU 2189072 C2

RU 2189072 C2

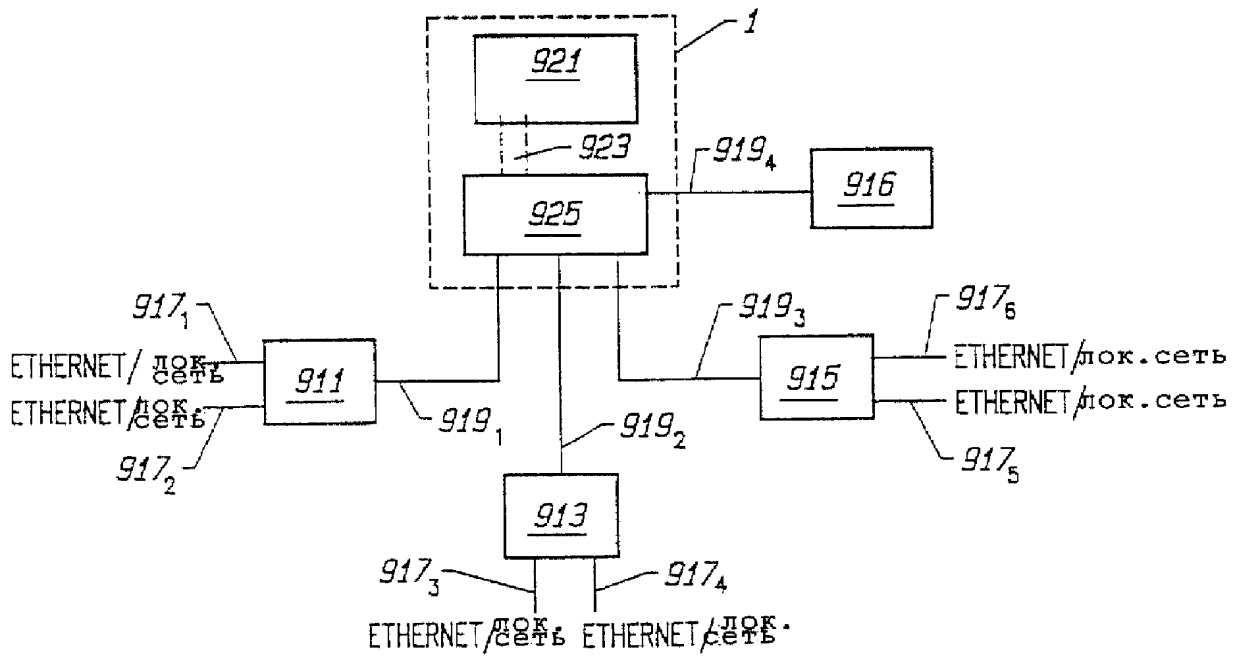
Переключающий сердечник	
Комплект микросхем	Комплект микросхем АТМ коммутатора MMC Networks ATMS 2000 (белая микросхема, серая микросхема, микросхемы MBUF, микросхемы PIF)
Память общих данных	Стандартные модули памяти
Счетчики пакетов	Стандартные счетчики
Микроконтроллерный комплекс	
ЦПБ	Intel 960CA/CF/HX
Динамическое ОЗУ	Стандартные модули динамического ОЗУ
ПЗУ	Стандартное ПЗУ
Флэш-память	Стандартная флэш-память
Контроллер динамич.ОЗУ	Стандартные микросхемы FPGA, ASIC и т.д.
Сдвоенный универсальный асинхронный приемопередающий порт	16552 DUART
Внешний таймер	Стандартный таймер
Подузел приемопередатчика	
Физический интерфейс	PMC-Sierra PM5346



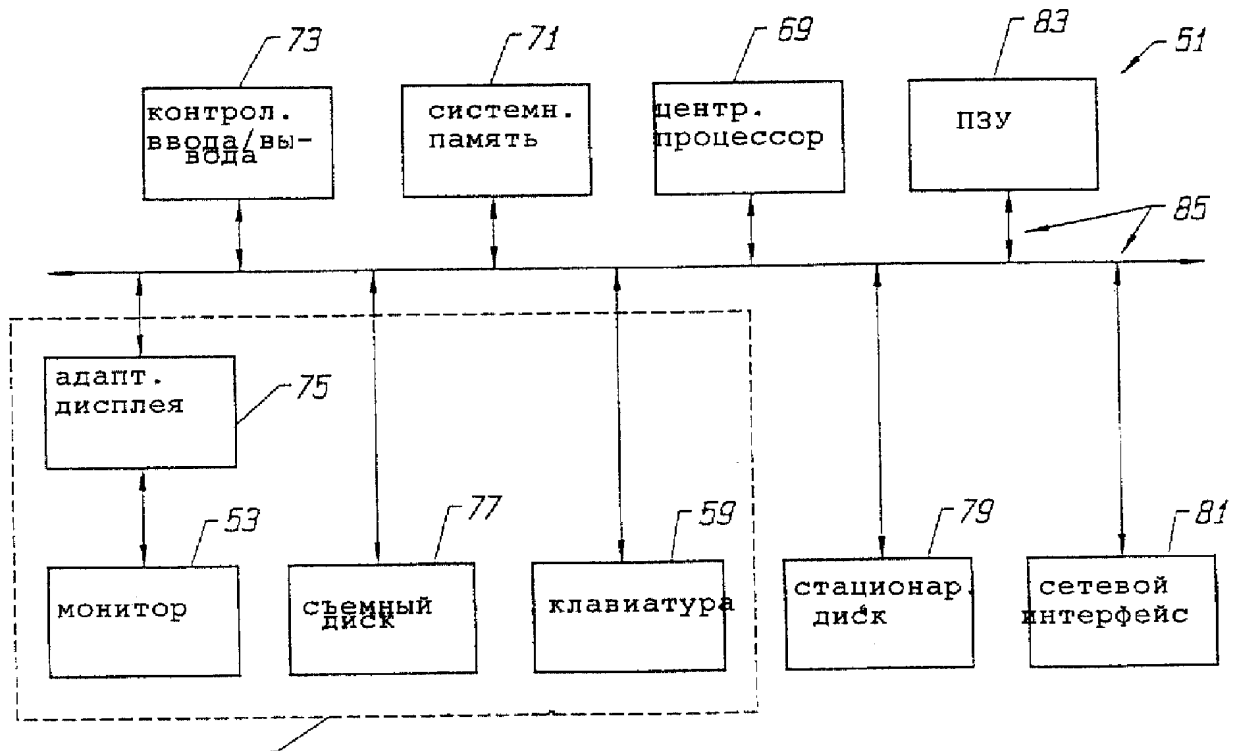
Фиг. 2а



Фиг. 2б

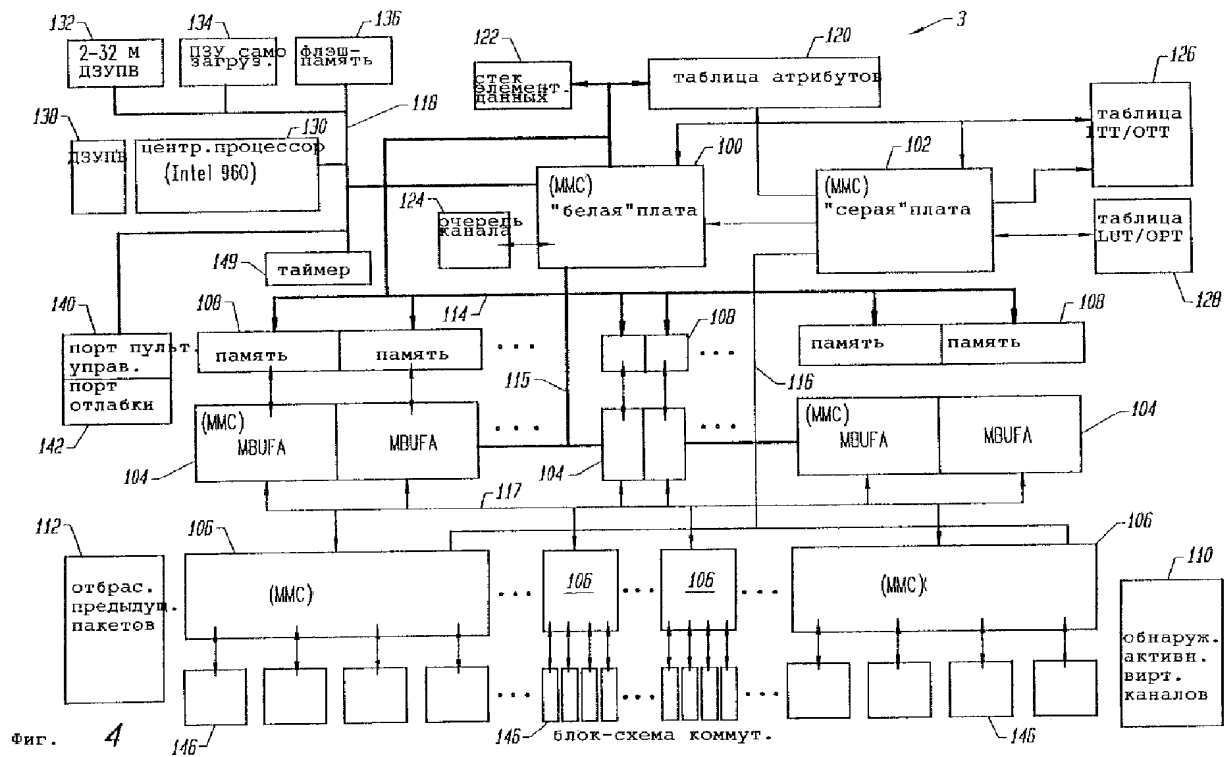


Фиг. 2С

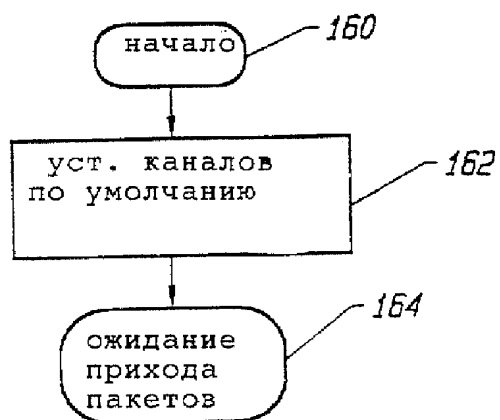


дополнит. средства
(для главного компьютера)

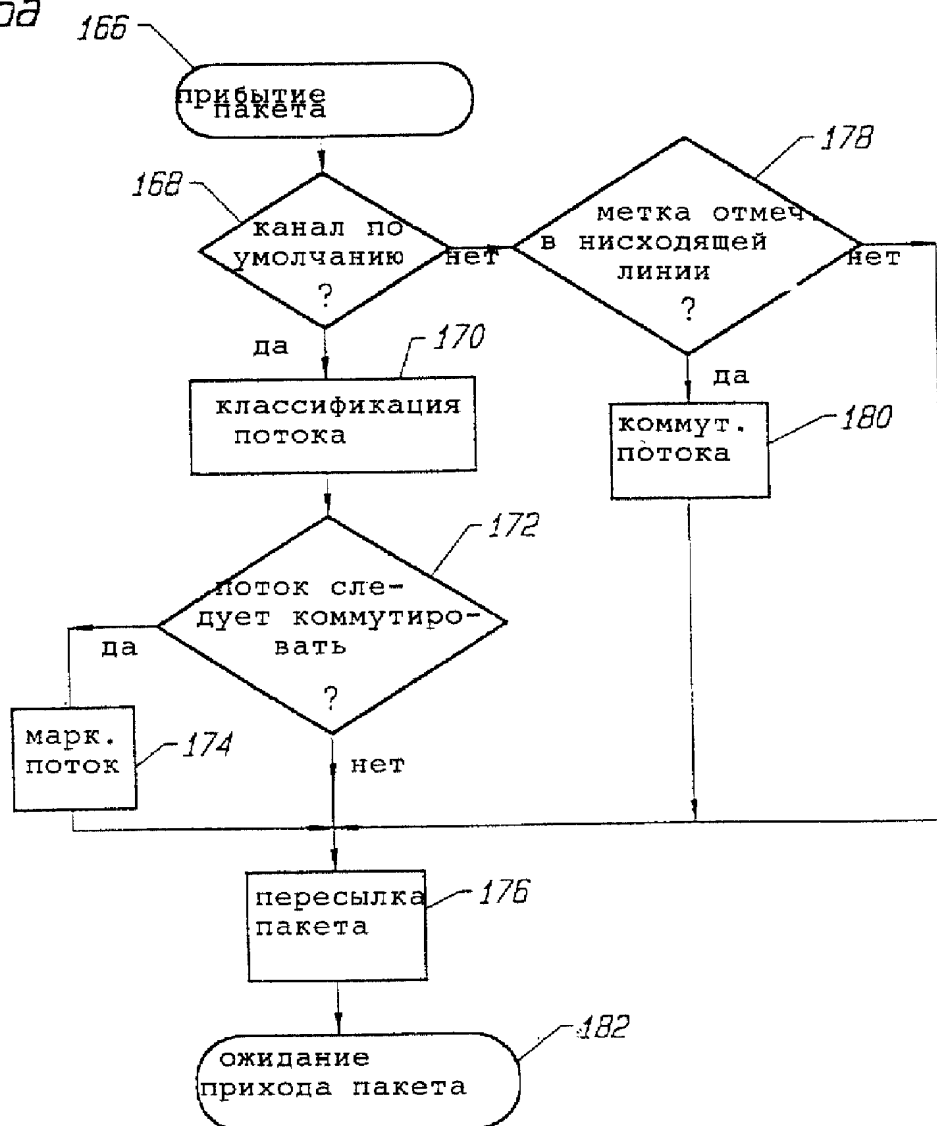
Фиг. 3



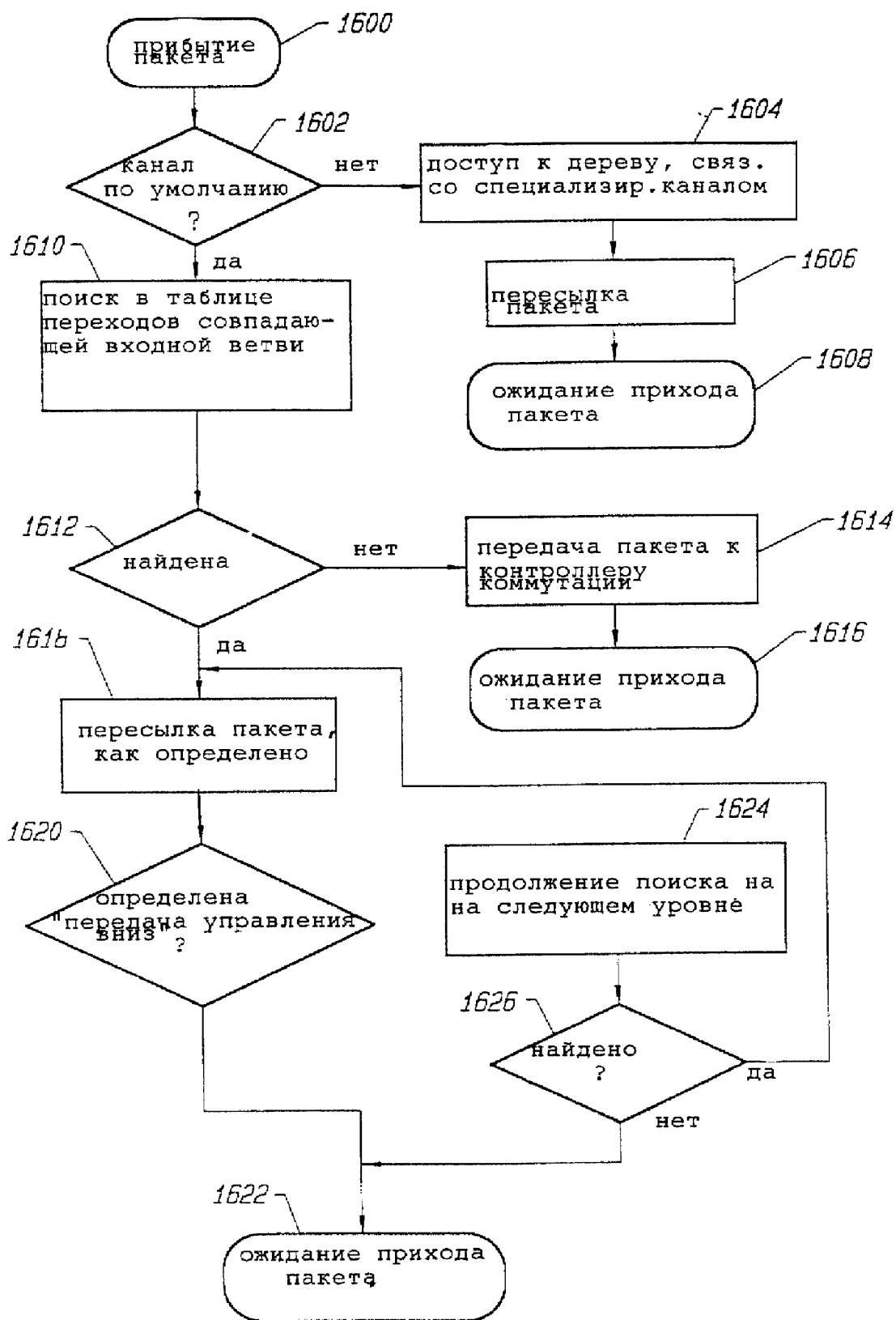
Фиг. 4



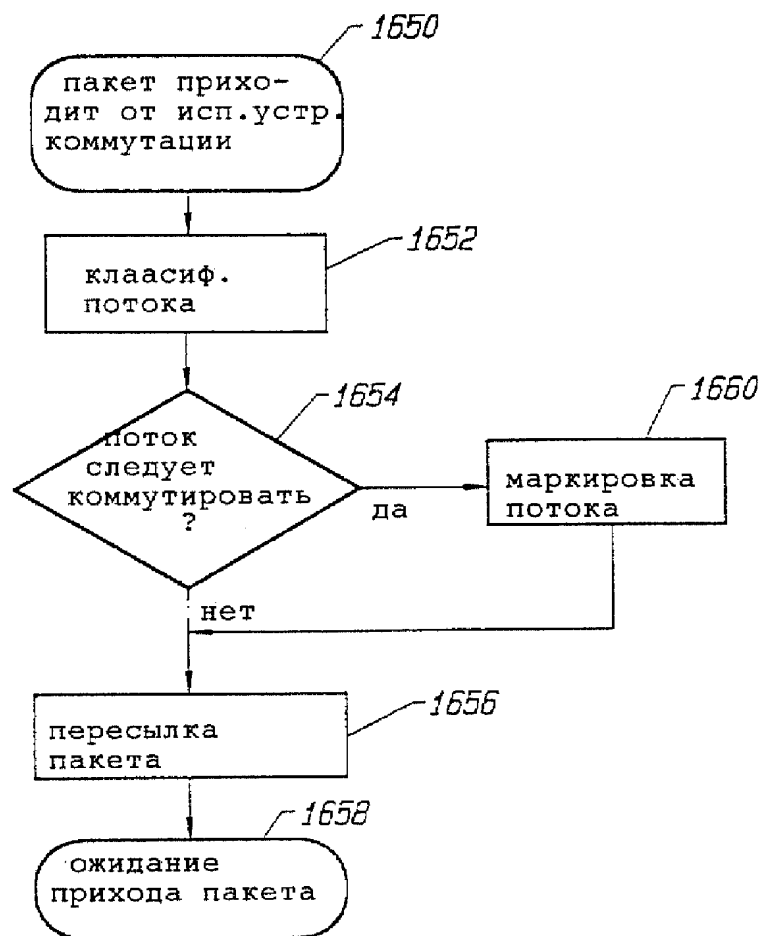
Фиг. 5а



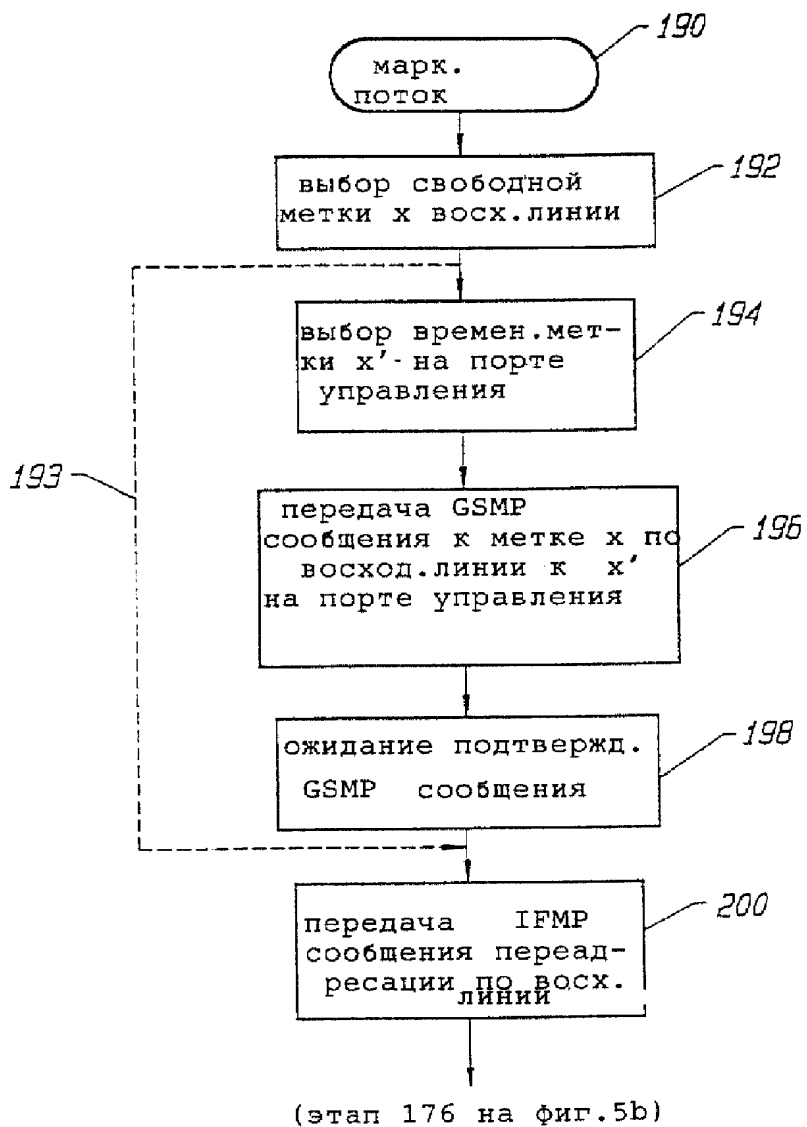
Фиг. 5б



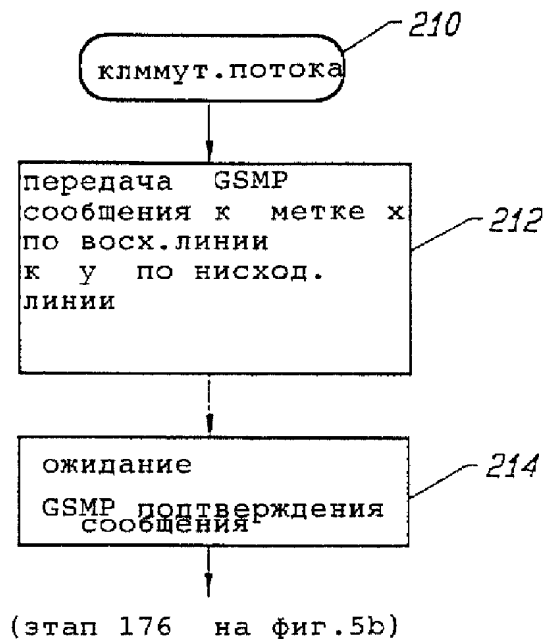
Фиг. 5с



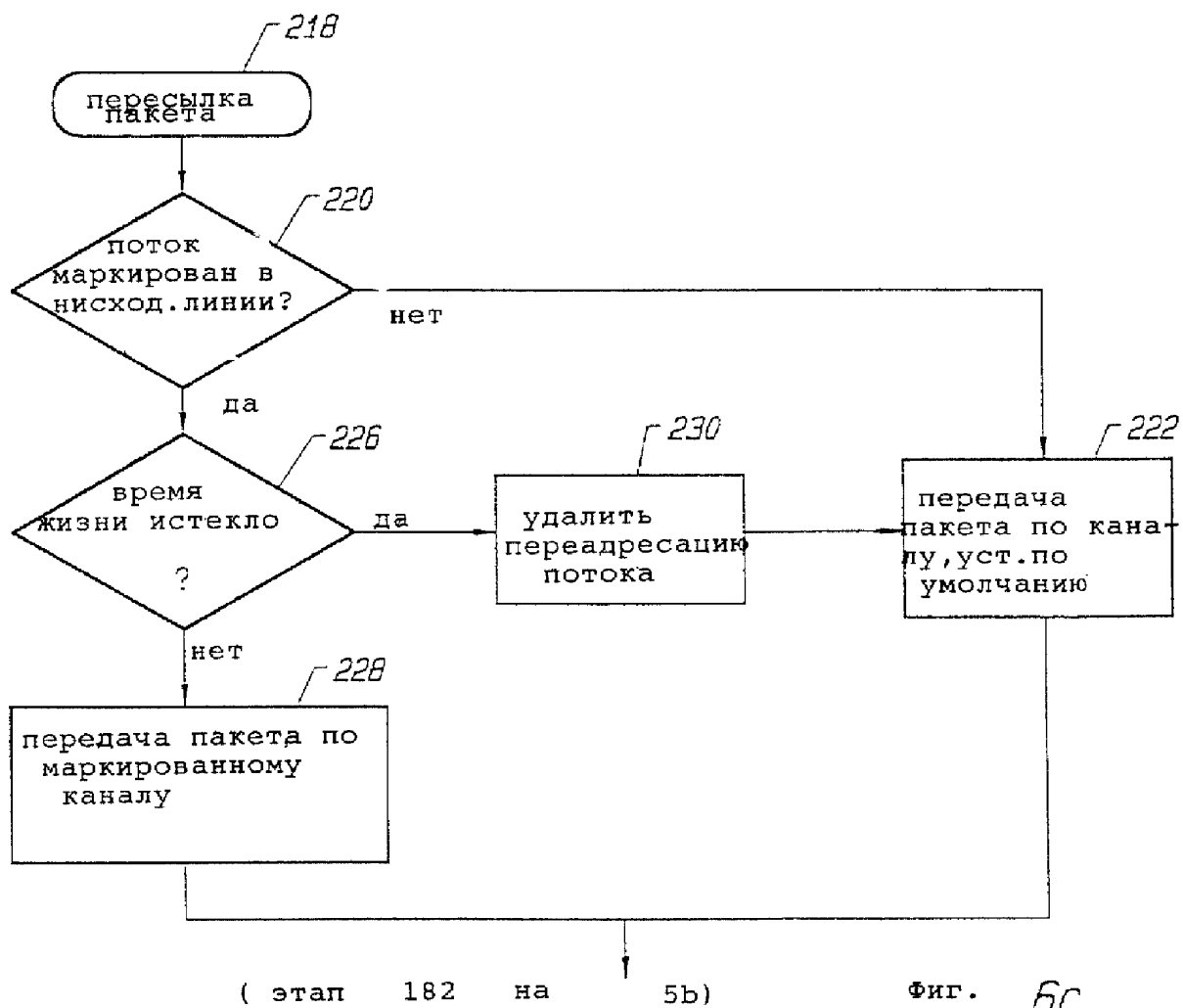
Фиг. 5d



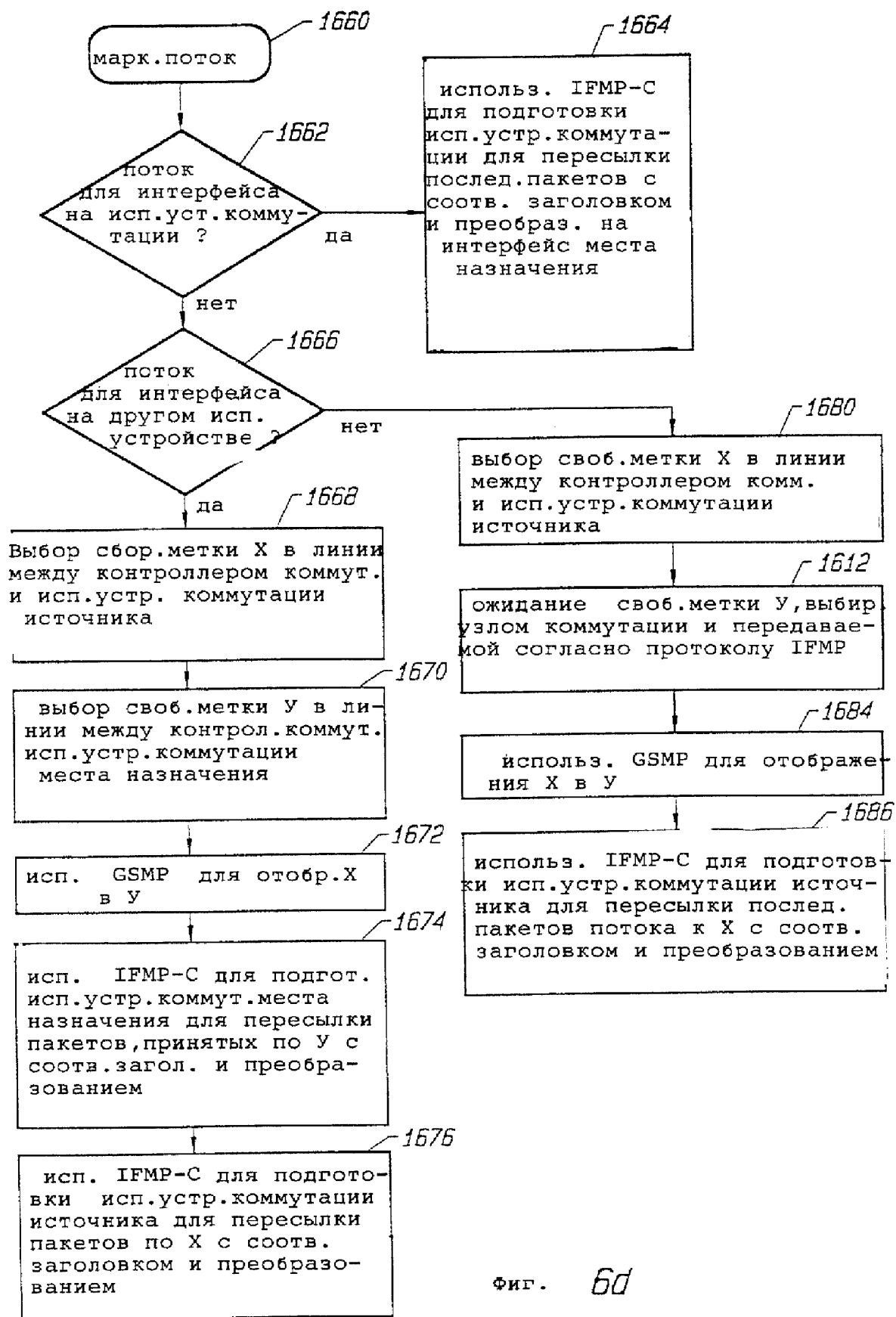
Фиг. 6а



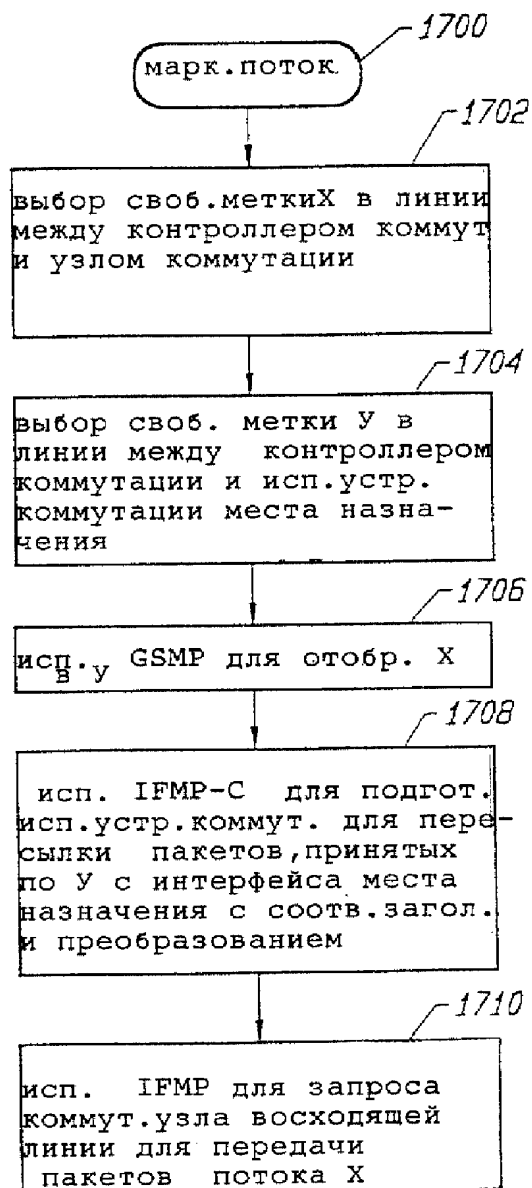
Фиг. 6b



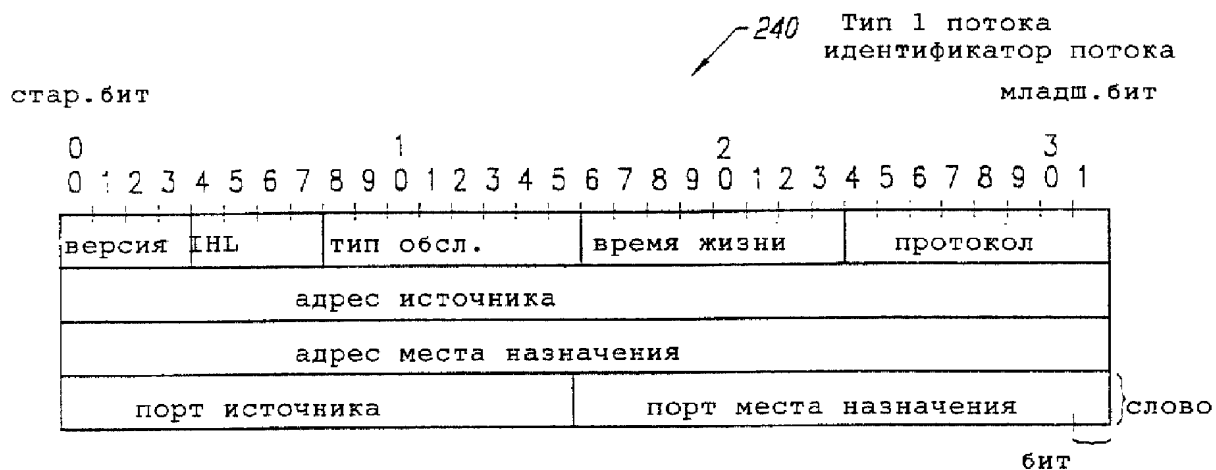
Фиг. 6c



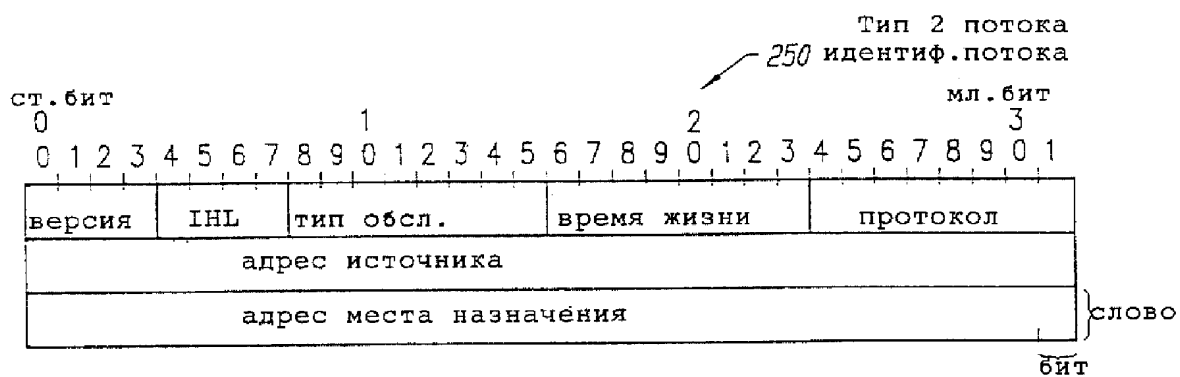
Фиг. 6d



Фиг. 6е



Фиг. 7a



Фиг. 7b

RU 2189072 C2

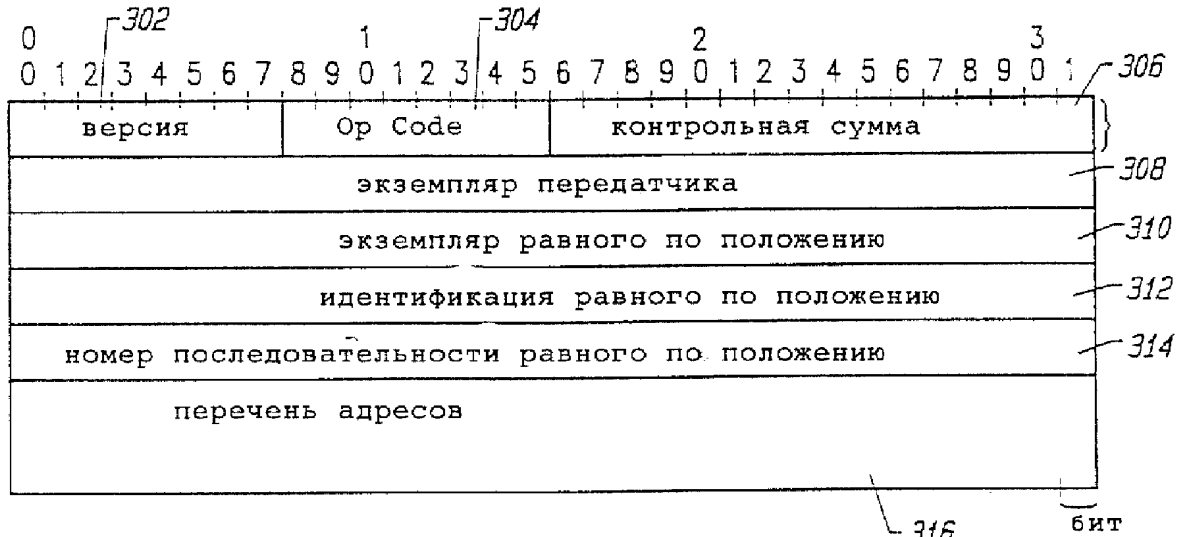
RU 2189072 C2

обобщенное сообщение
 протокола близости
 IFMP

300

ст. бит

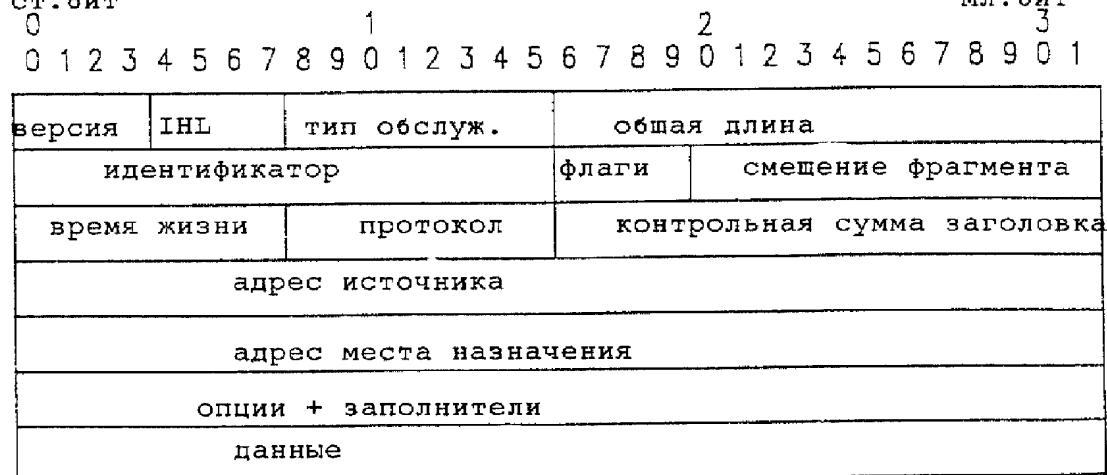
мл. бит



Фиг. 8a

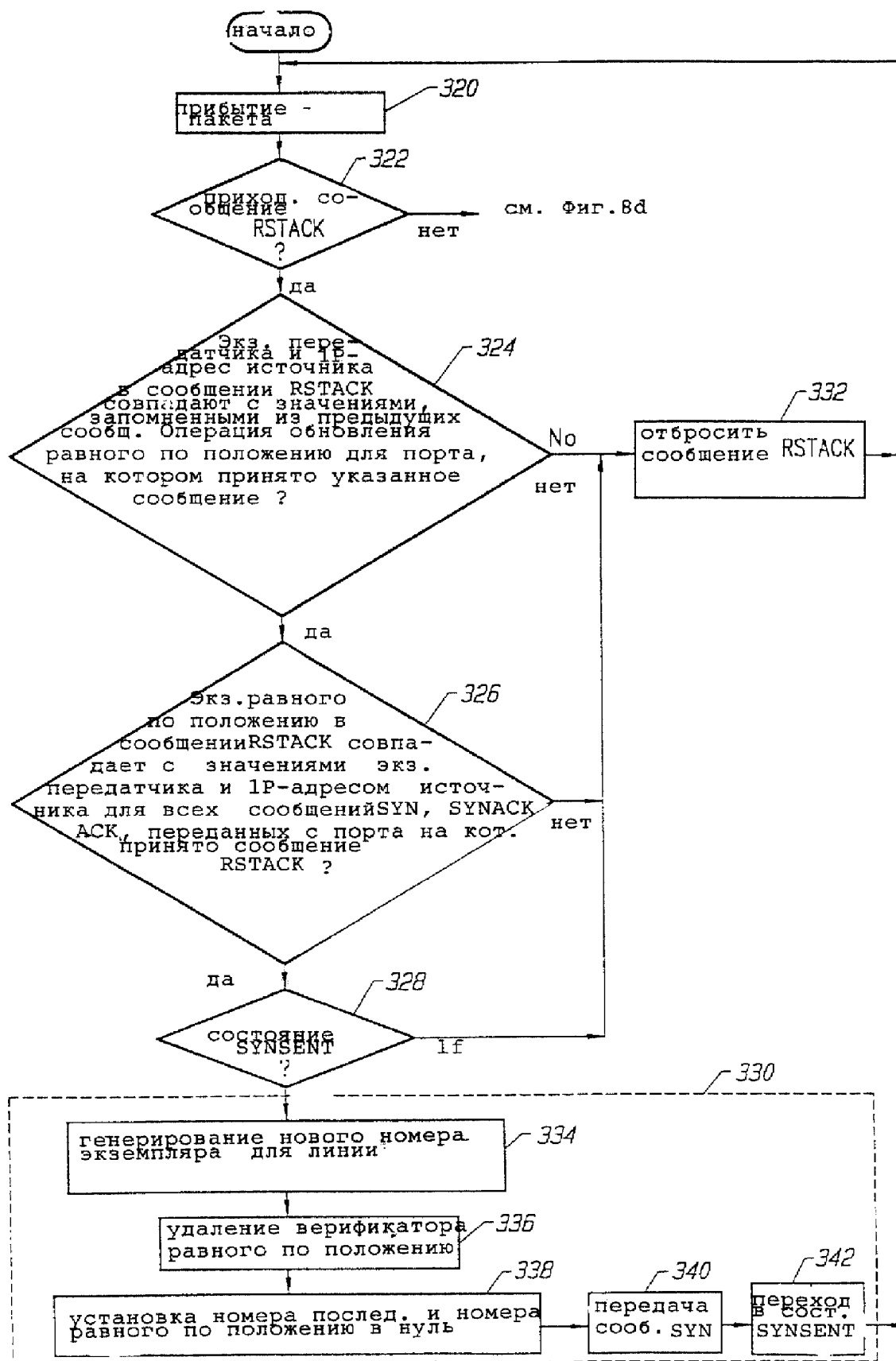
ст. бит

мл. бит

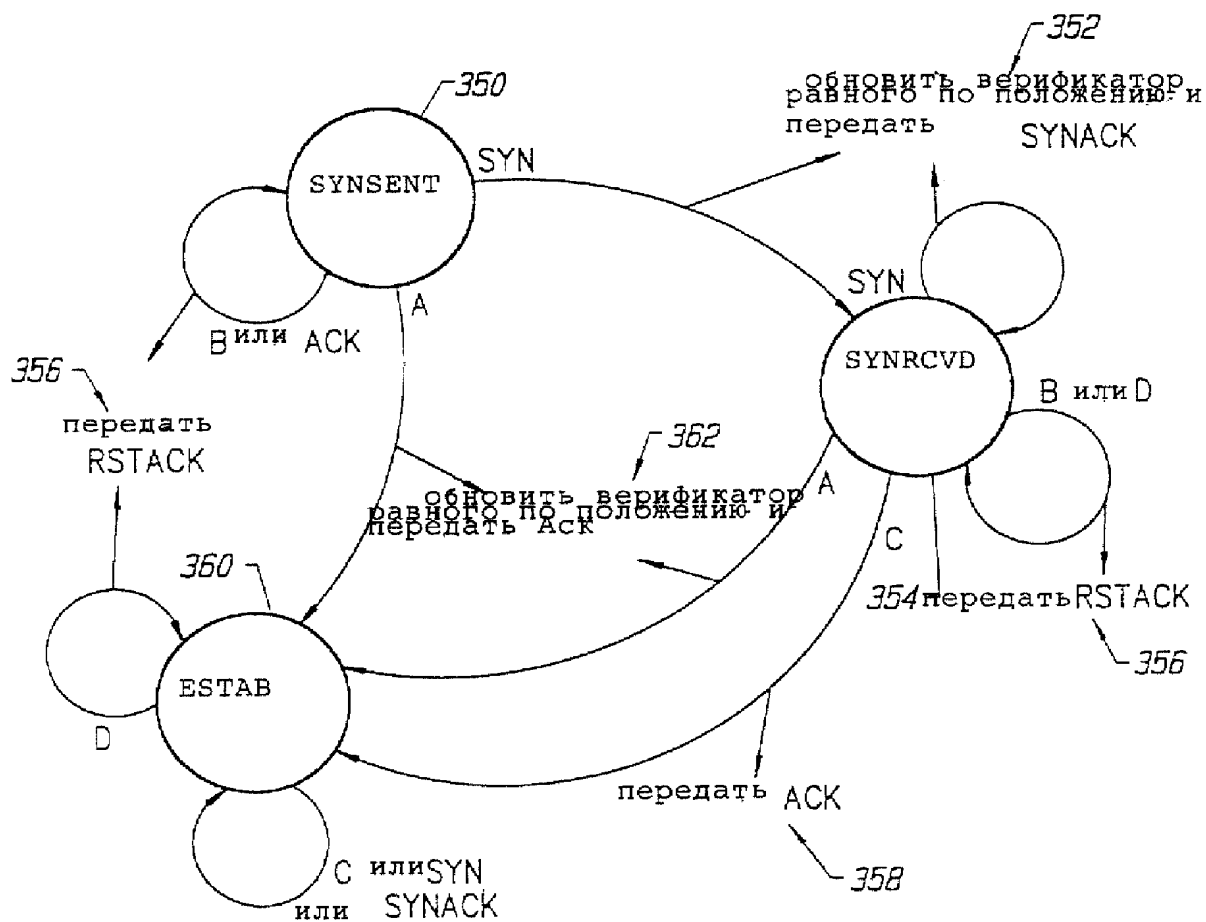


обобщ. IP пакет (IPv4)

Фиг. 8b



Фиг. 8с



Условия:

A: SYNACK AND %C

B: SYNACK AND NOT(%C)

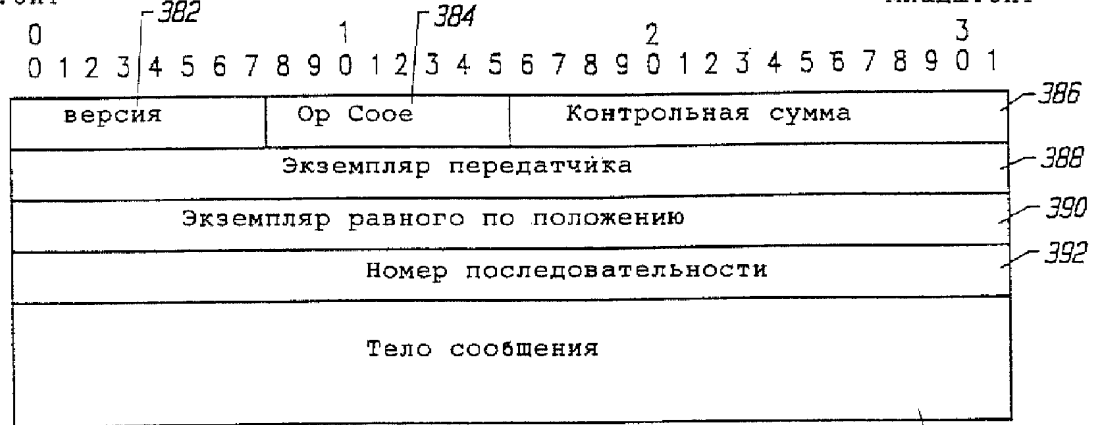
C: ACK AND %B AND %C

D: ACK AND NOT (%B AND %C)

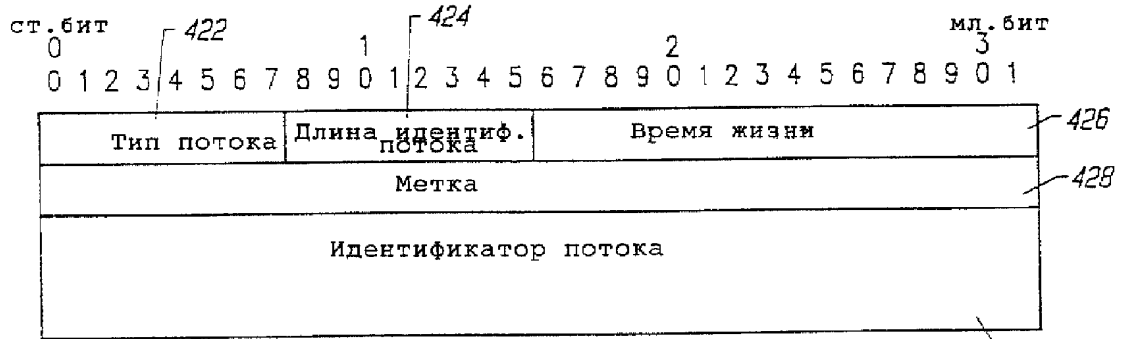
ФИГ. 8d

старш. бит

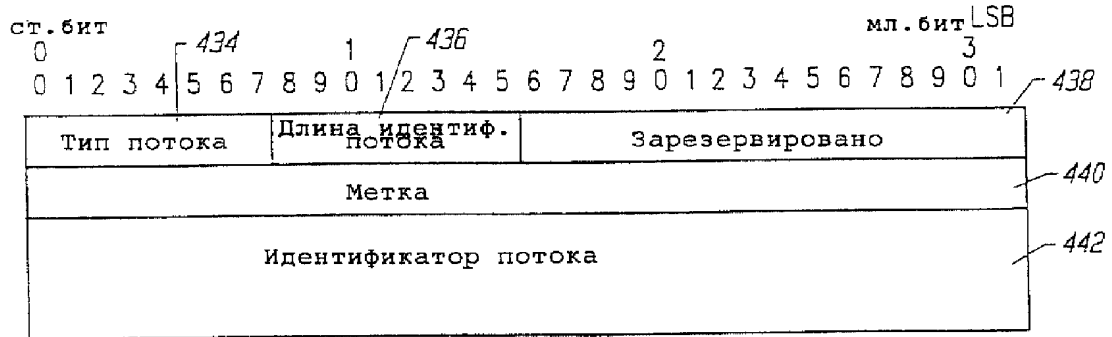
младш. бит



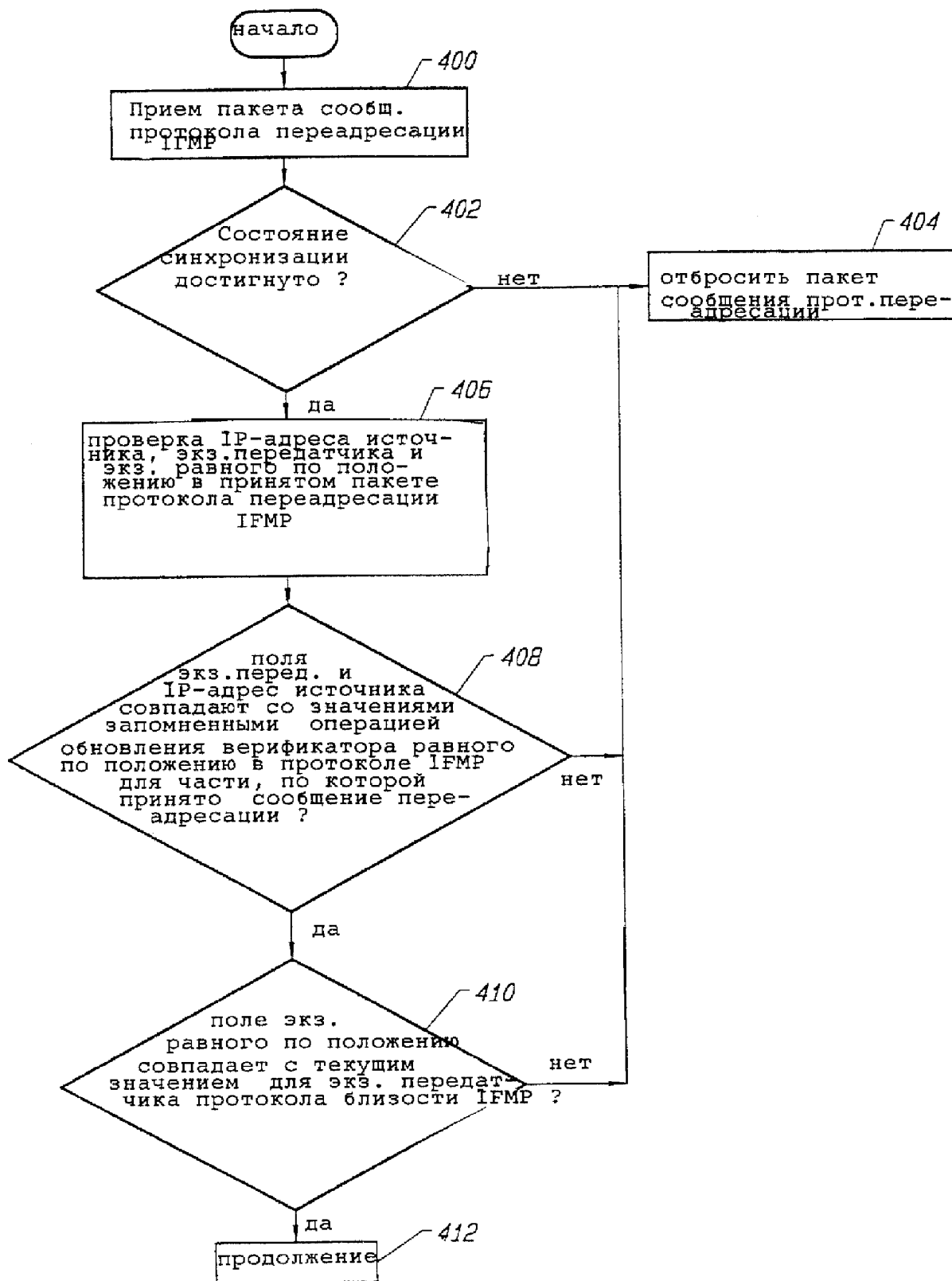
Фиг. 9a

элемент сообщения
переадресовки

Фиг. 9c

элемент сообщения
восстановления

Фиг. 9d

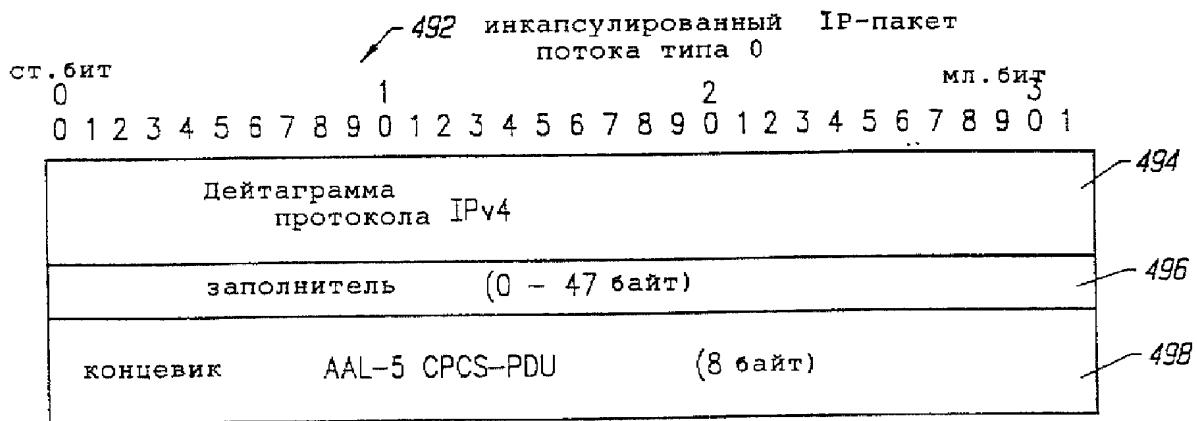


Фиг. 9b

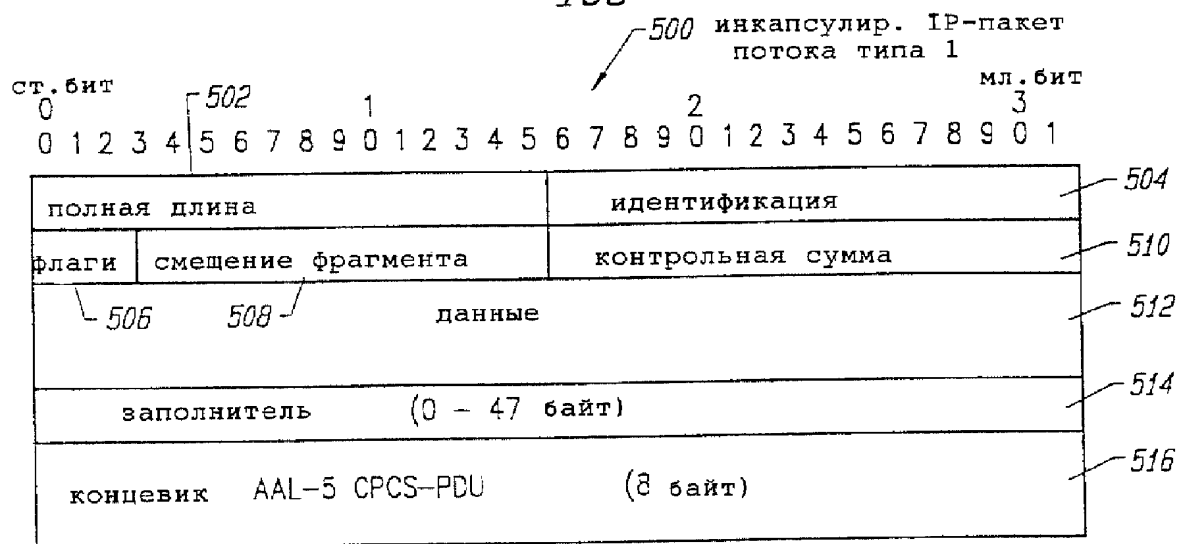


Фиг. 10а

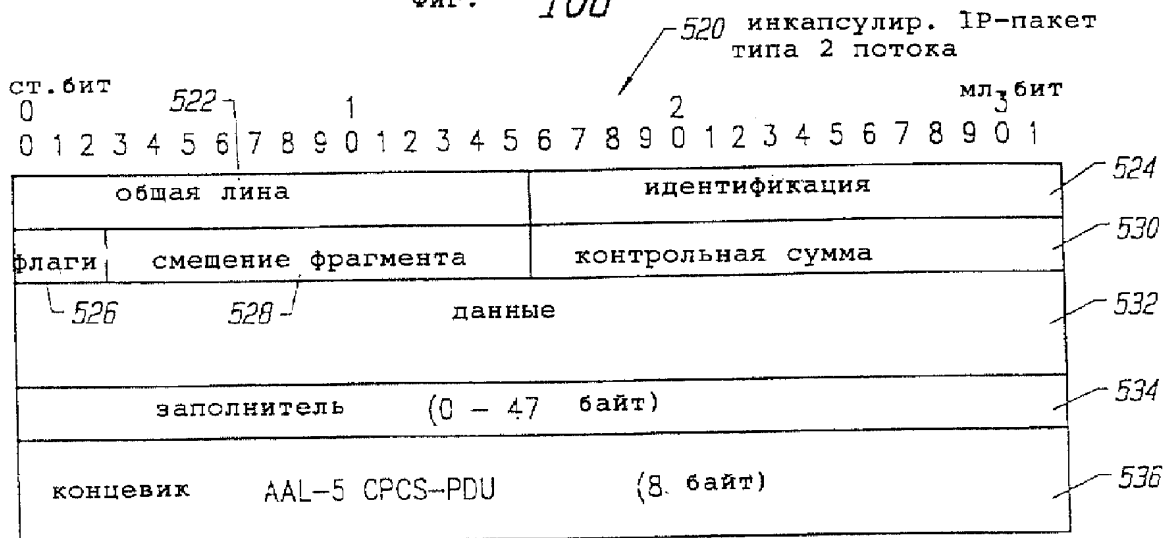
ФИГ. 10b



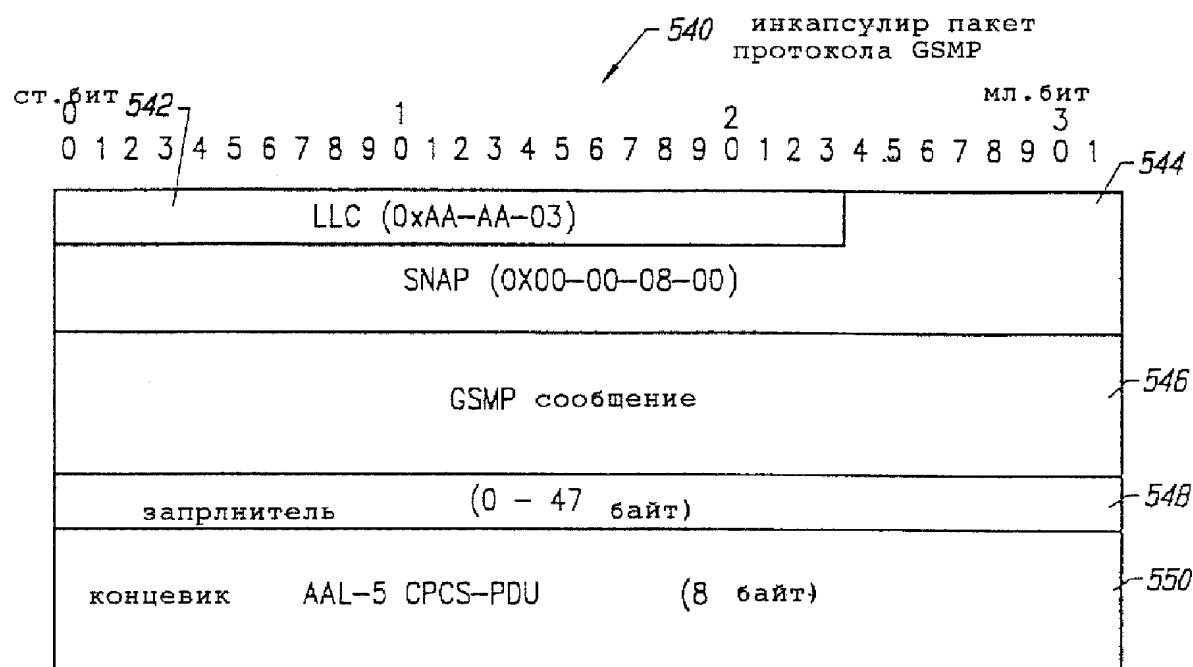
Фиг. 10c



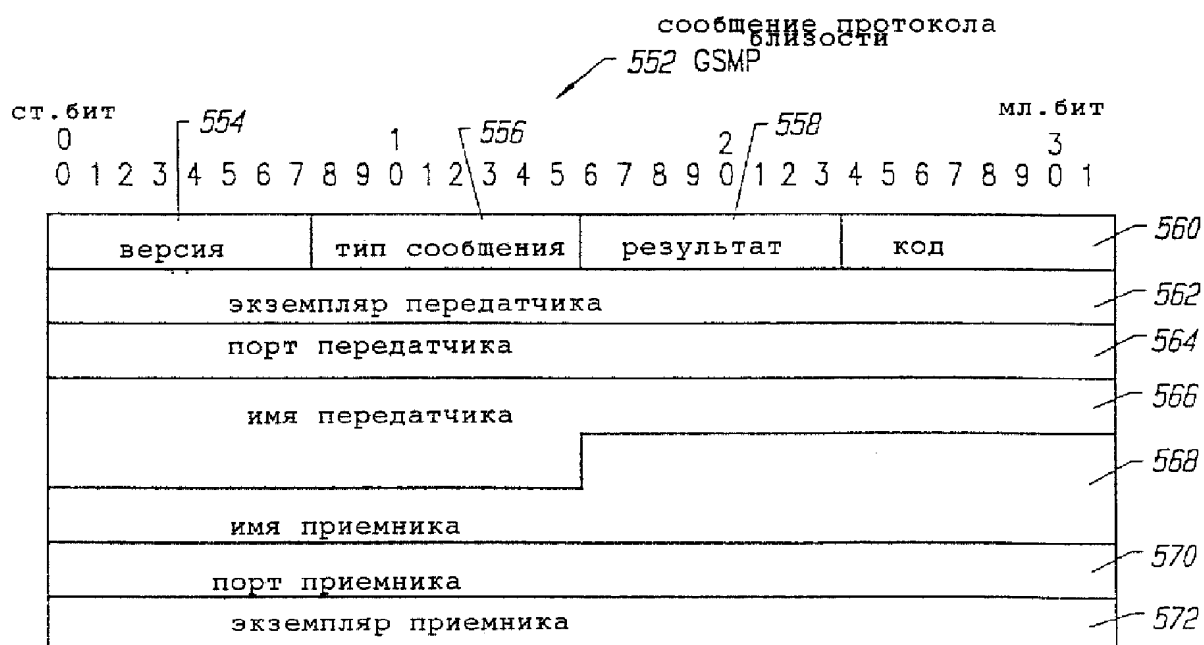
Фиг. 10d



Фиг. 10e



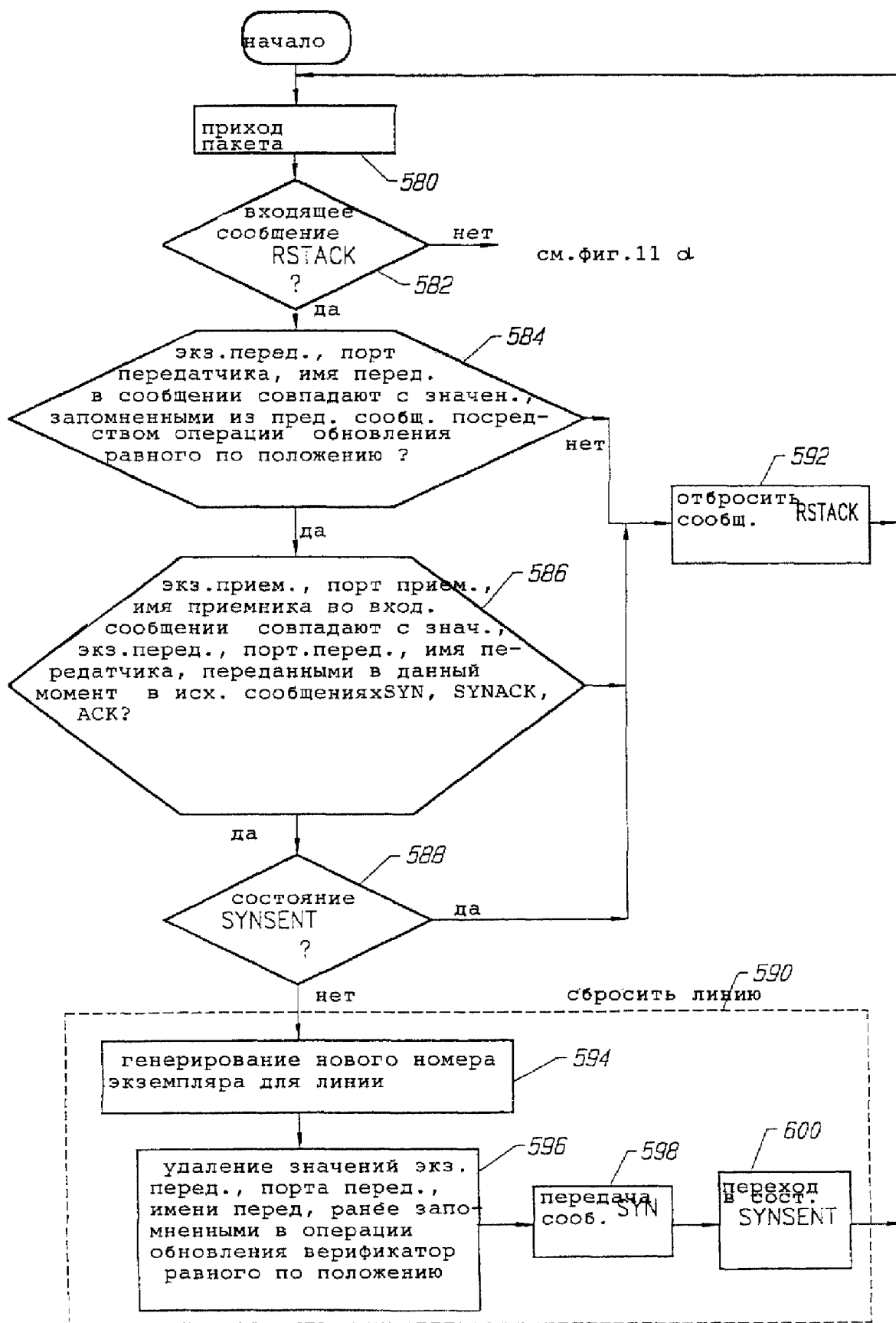
Фиг. 11a



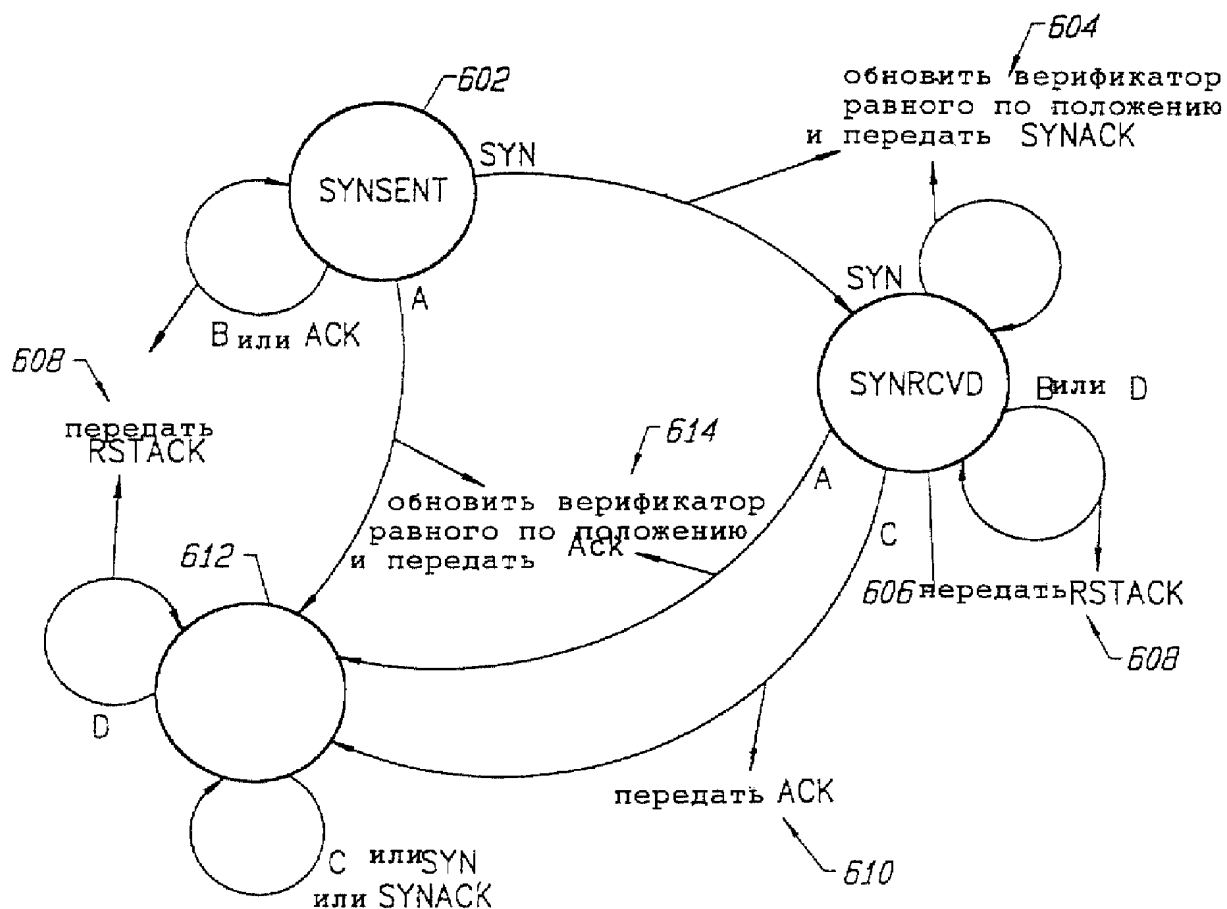
Фиг. 11b

RU 2189072 C2

RU 2189072 C2



Фиг. 11c



Условия:

A: SYNACK AND %C

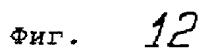
B: SYNACK AND NOT(%C)

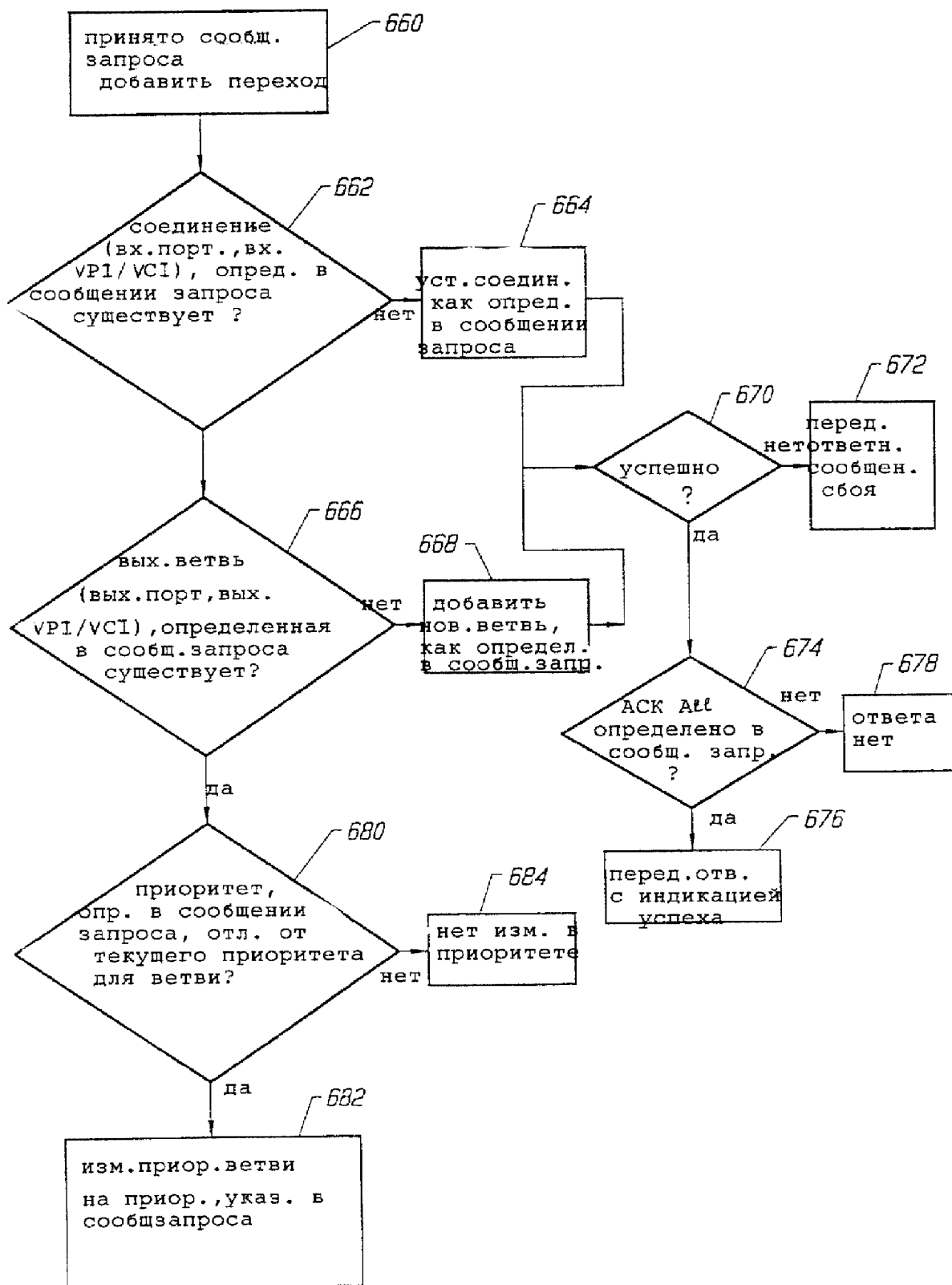
C: ACK AND %B AND %C

D: ACK AND NOT(%B AND %C)

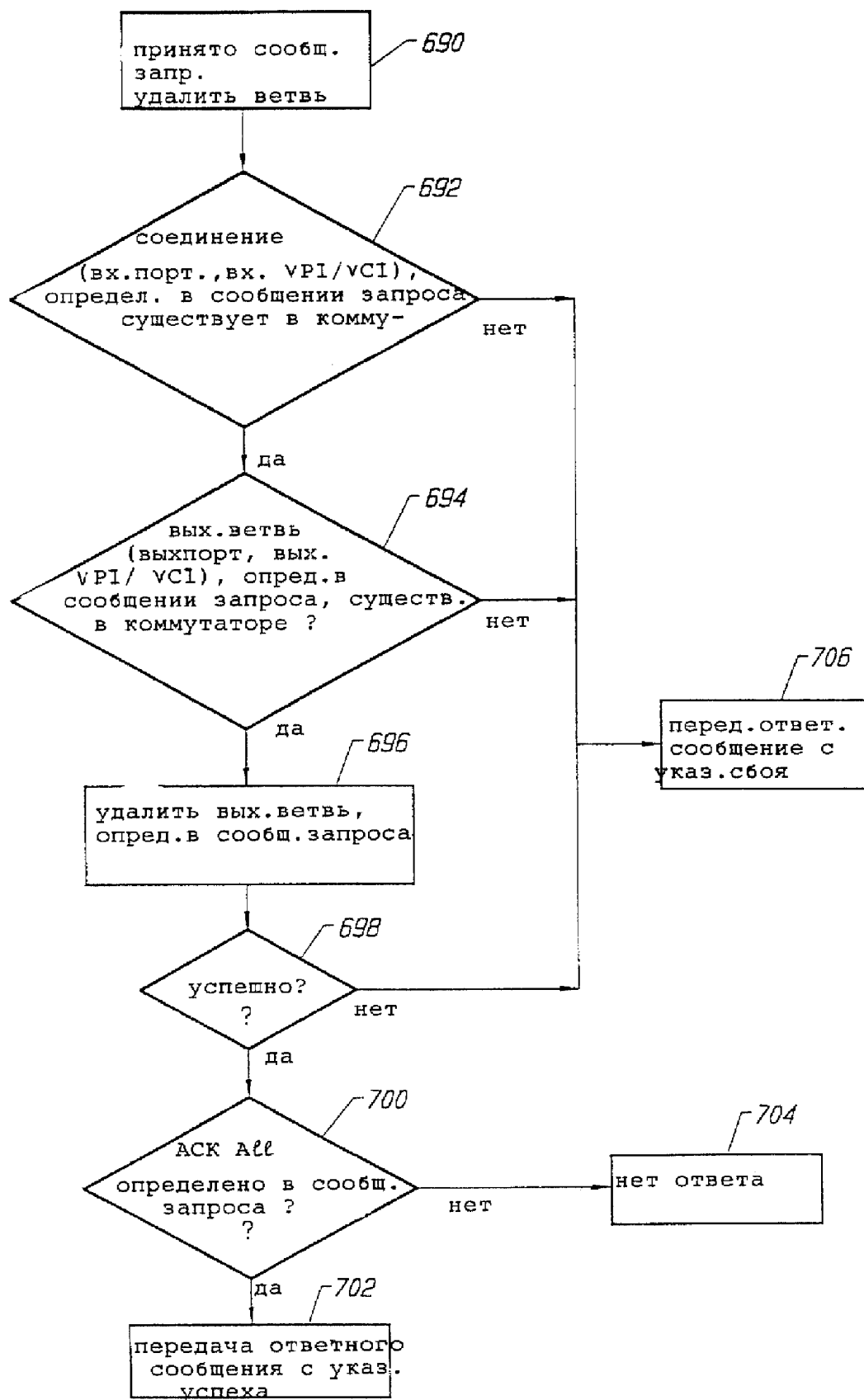
ФИГ. 11d

820

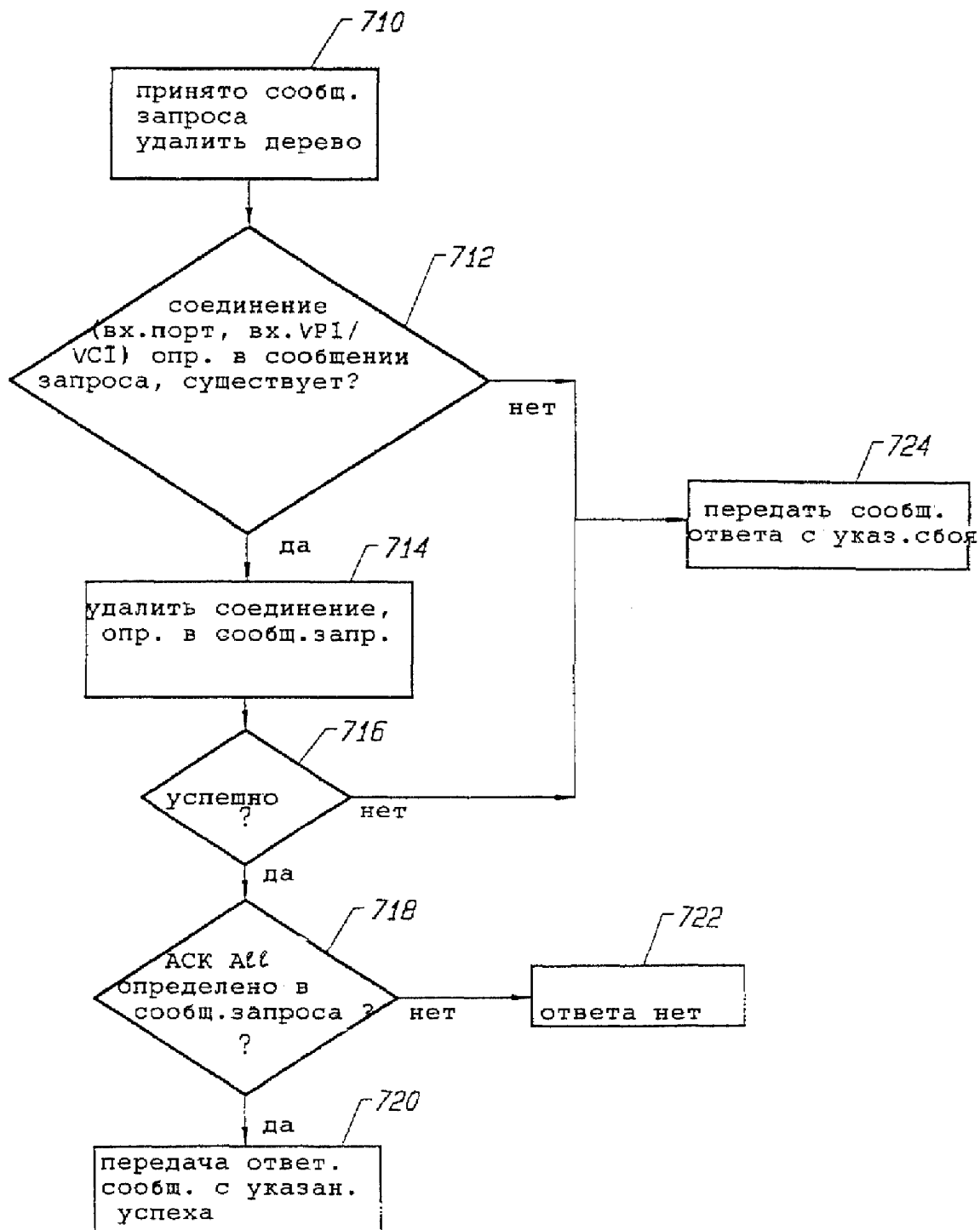




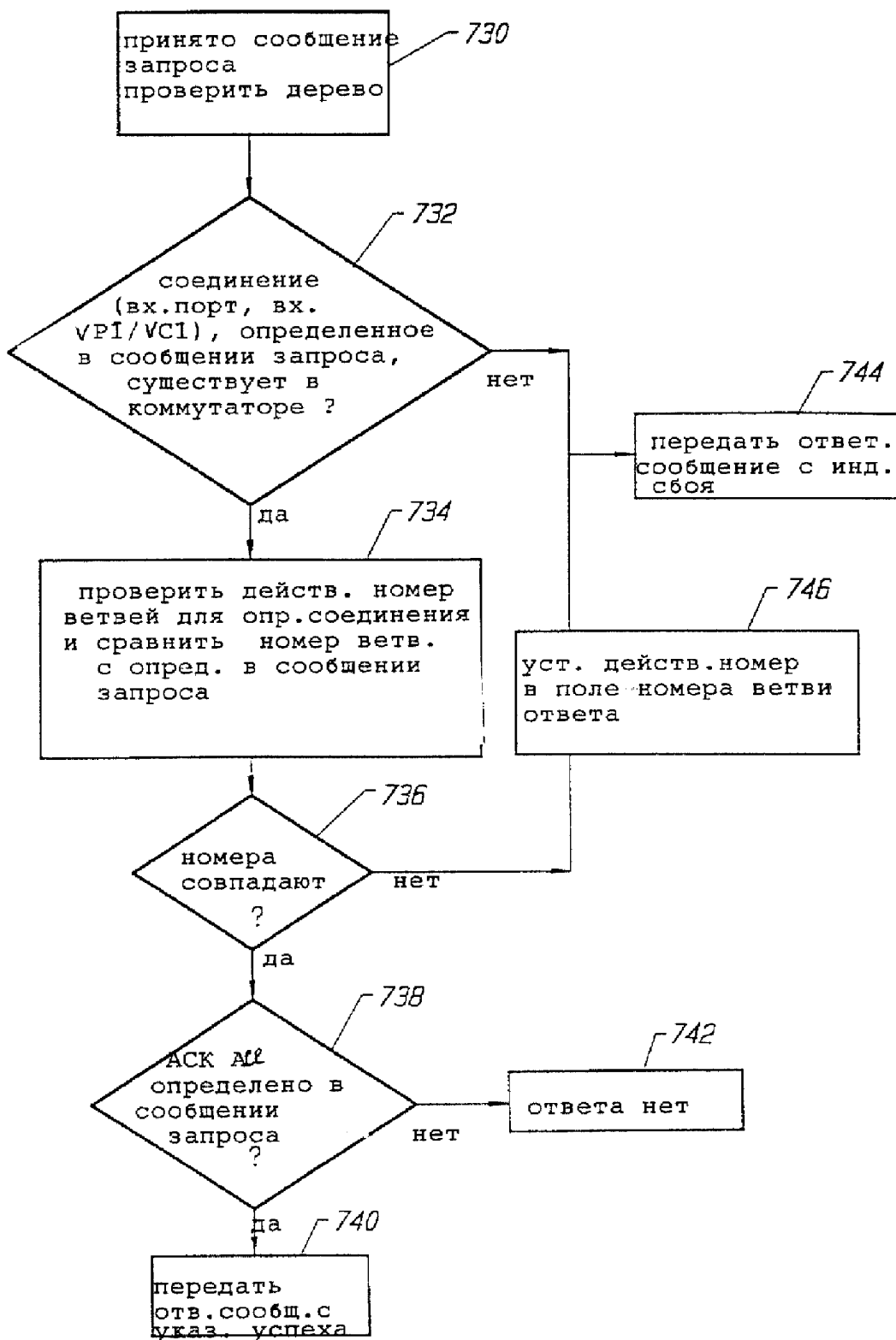
Фиг. 13а



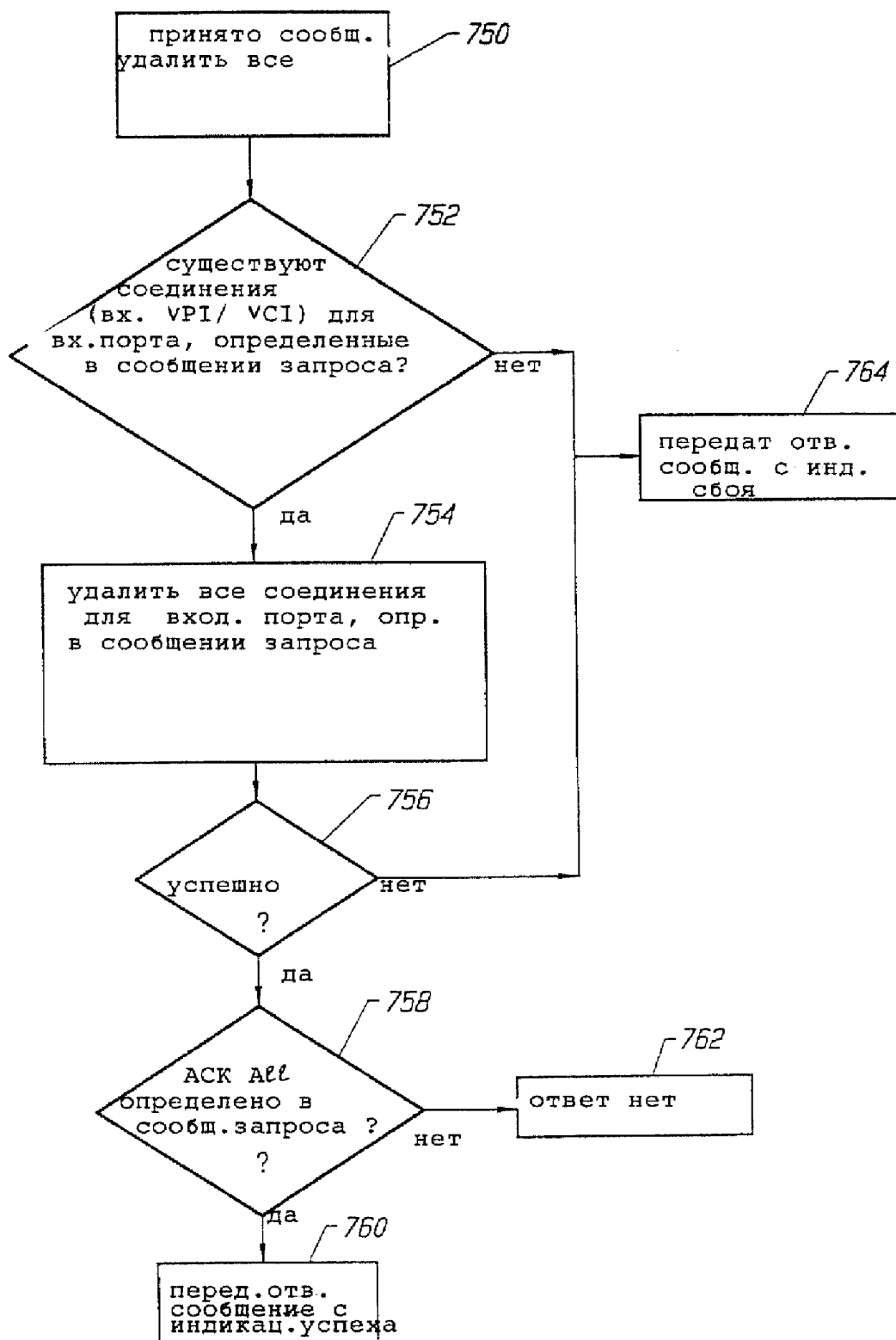
Фиг. 13b



Фиг. 13c

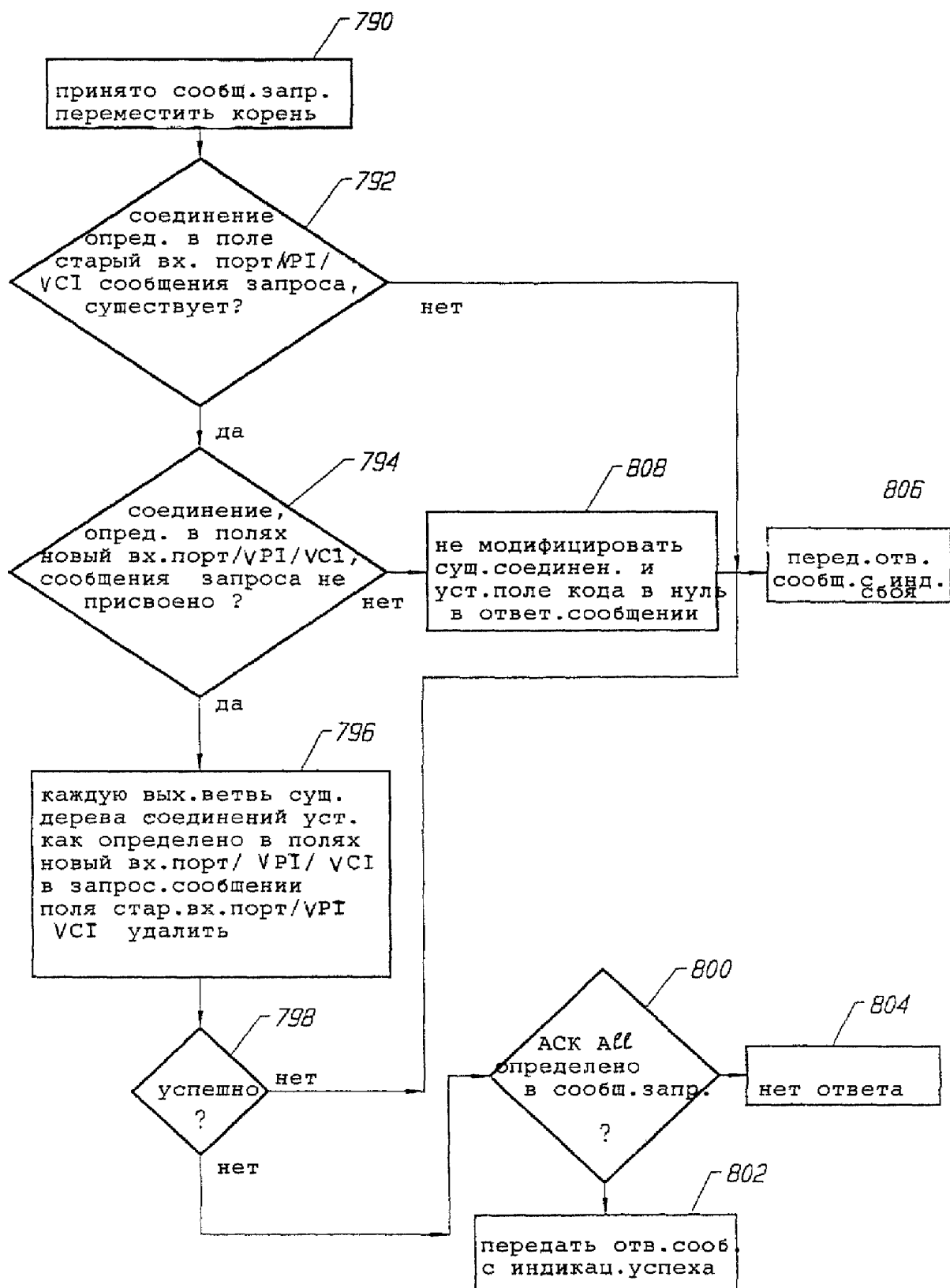


Фиг. 13d



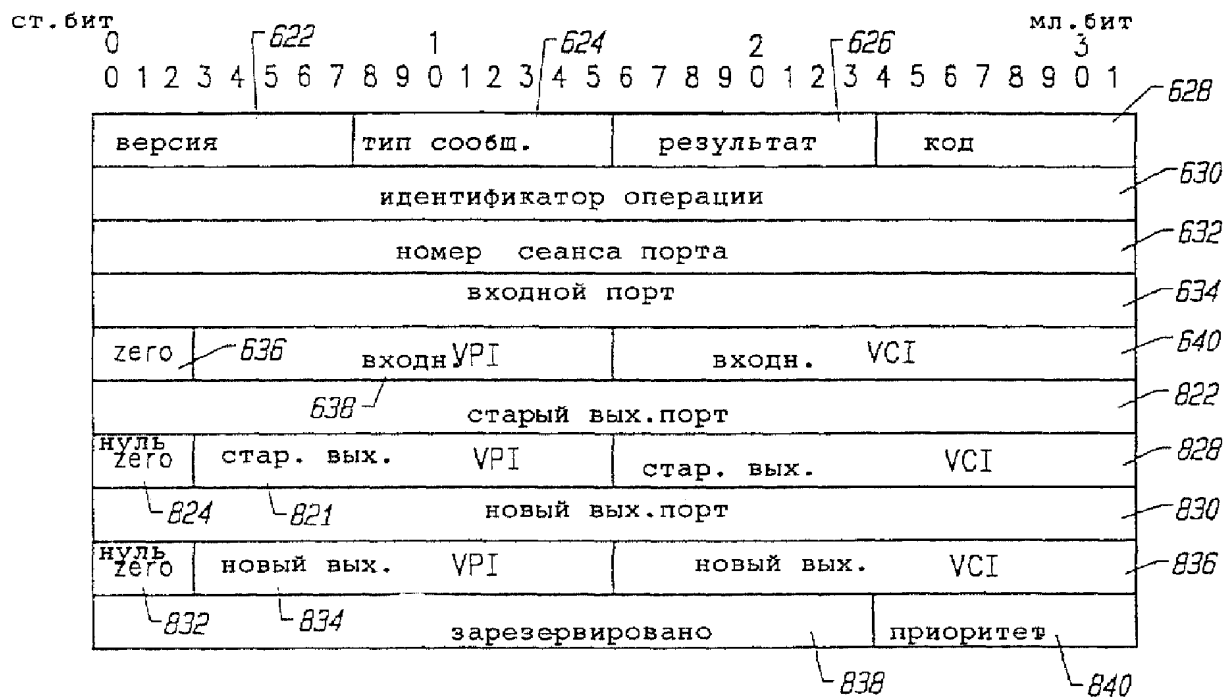
Фиг. 13e





Фиг.13g

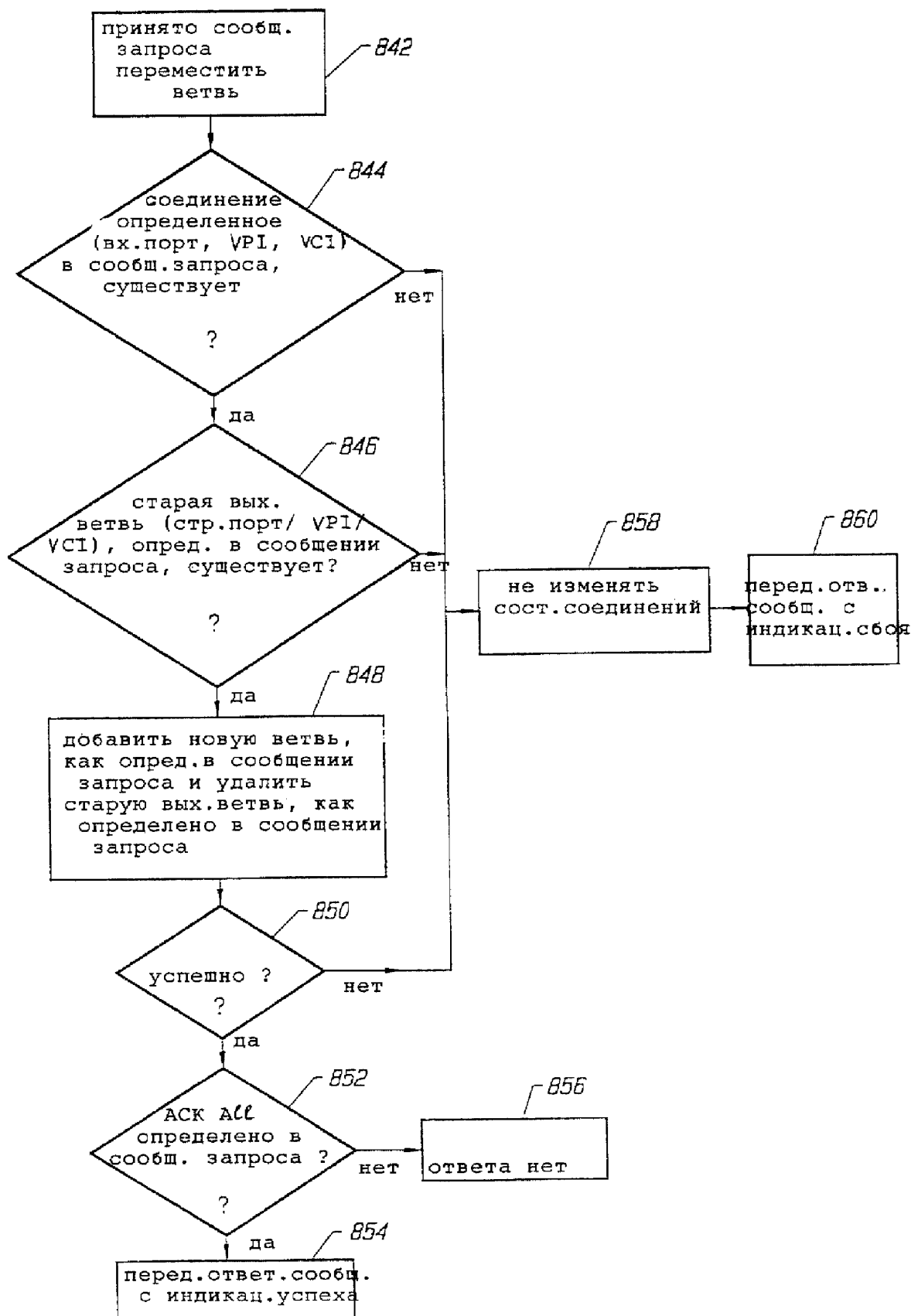
620 GSMP



Фиг. 13h

RU 2189072 C2

RU 2189072 C2

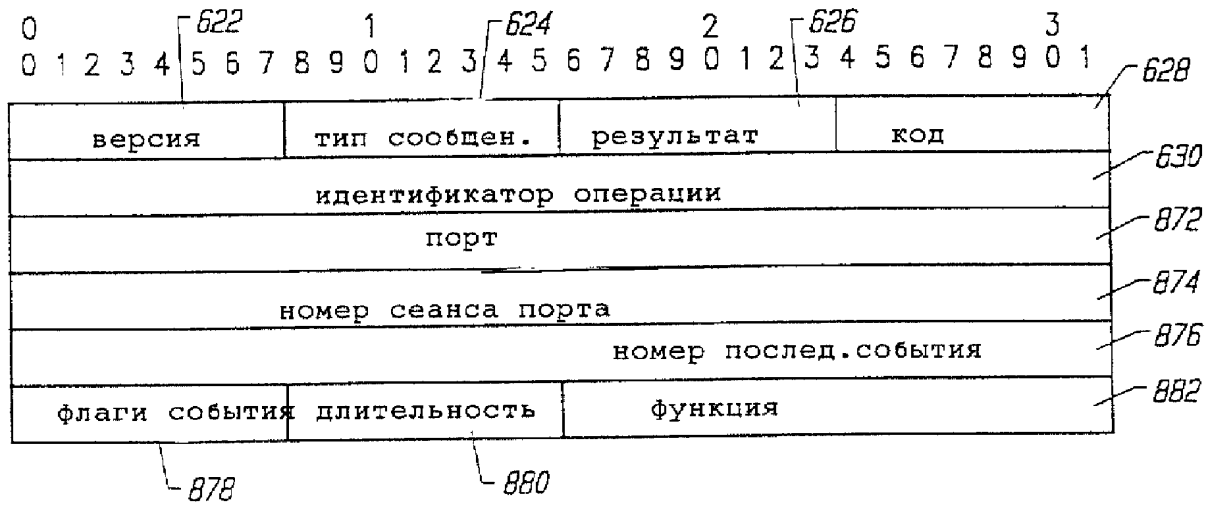


фиг. 13i

870 GSMP

ст. бит

мл. бит

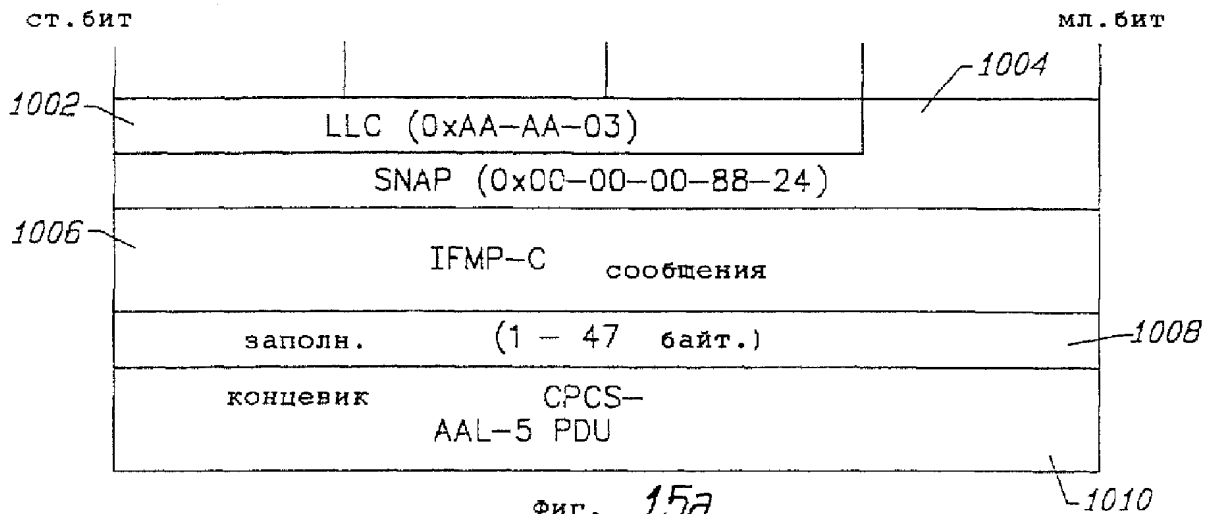


Фиг. 14

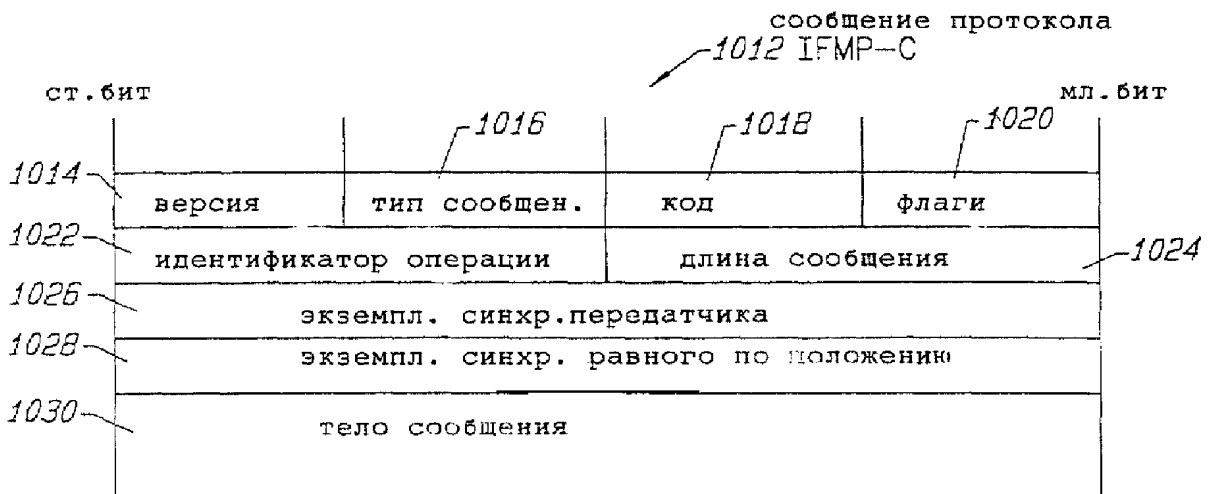
RU 2189072 C2

RU 2189072 C2

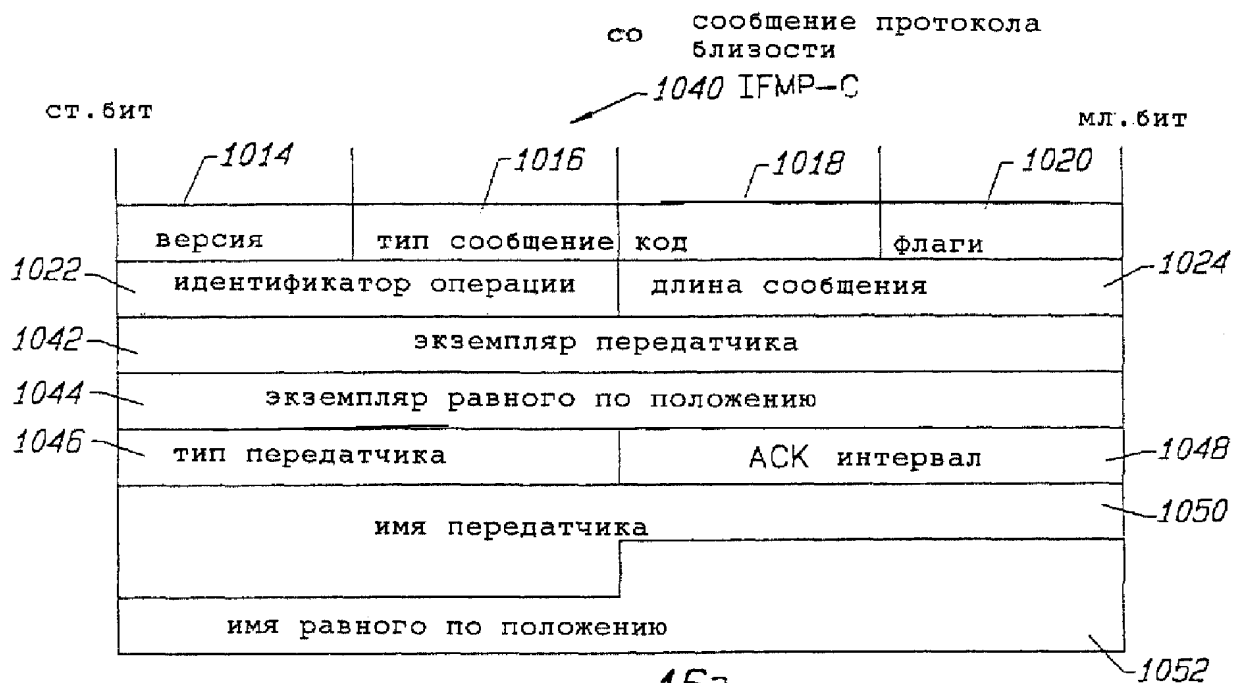
1000 инкапсулированный
IFMP-C пакет



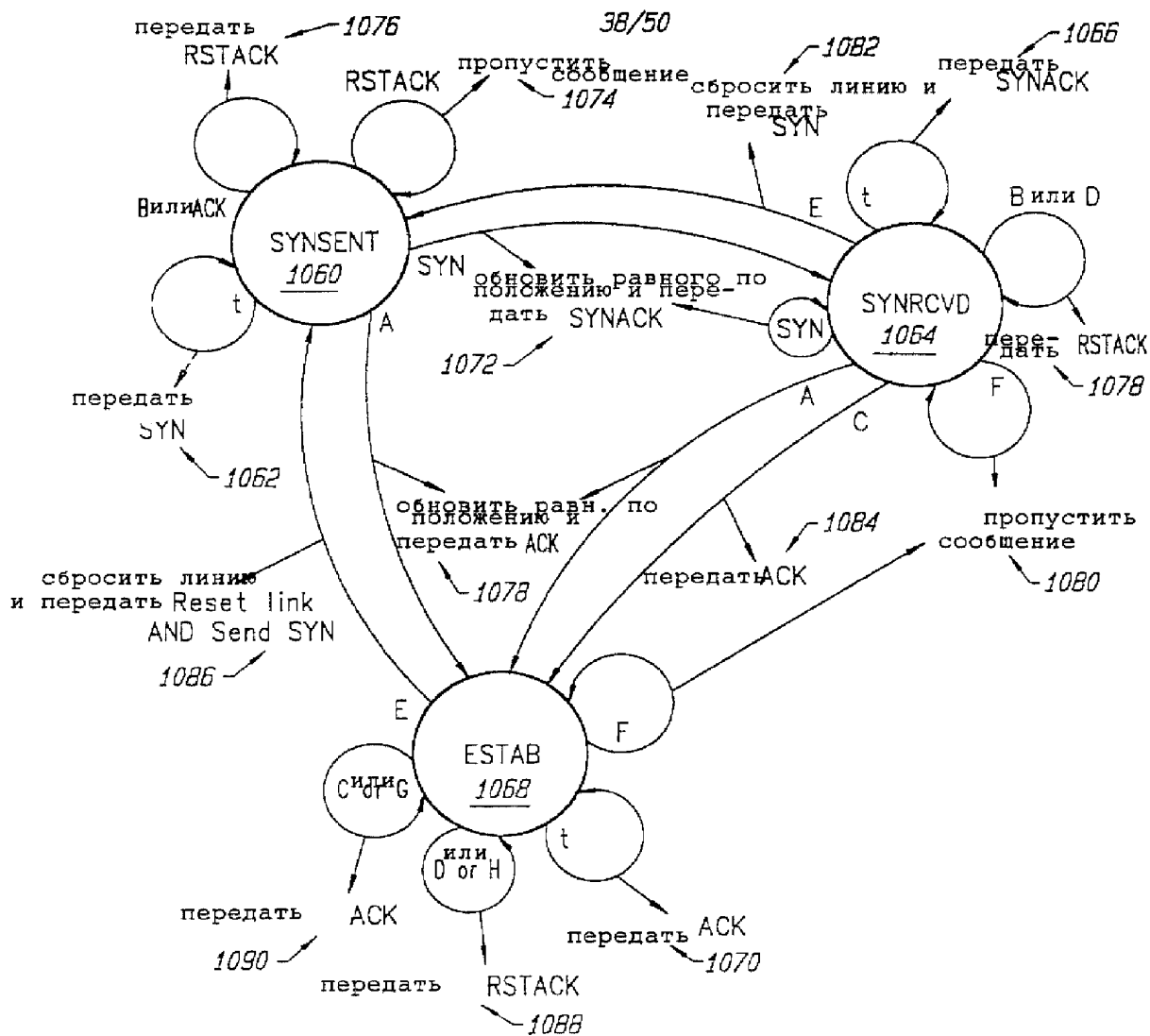
Фиг. 15a



Фиг. 15b

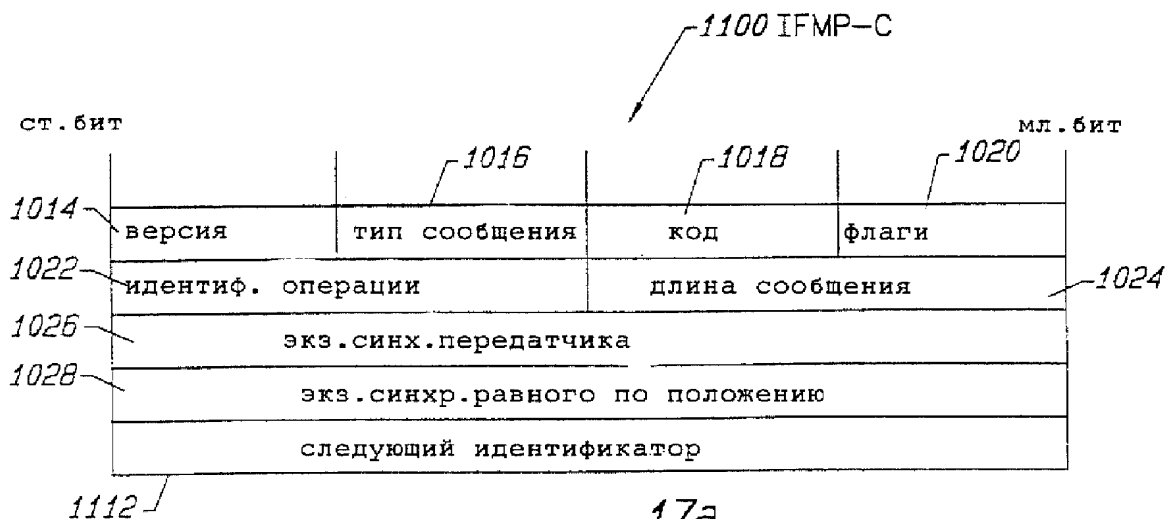


Фиг. 16а



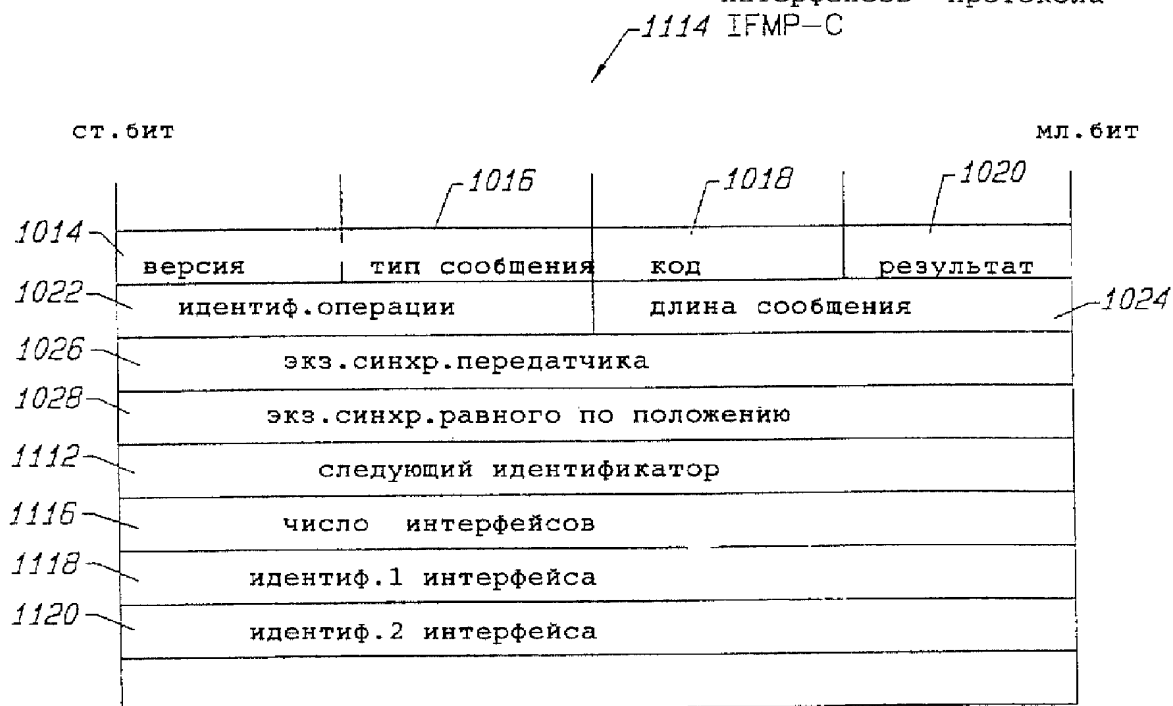
- A: SYNACK AND %X
- B: SYNACK AND NOT(%X)
- C: ACK AND %X AND %Y
- D: ACK AND NOT(%X AND %Y)
- E: RSTACK AND %X AND %Y
- F: RSTACK AND NOT(%X AND %Y)
- G: (SYN OR SYNACK) AND NOT (%X)
- H: (SYN OR SYNACK) AND NOT (%X)

Фиг. 16b



фиг. 17a

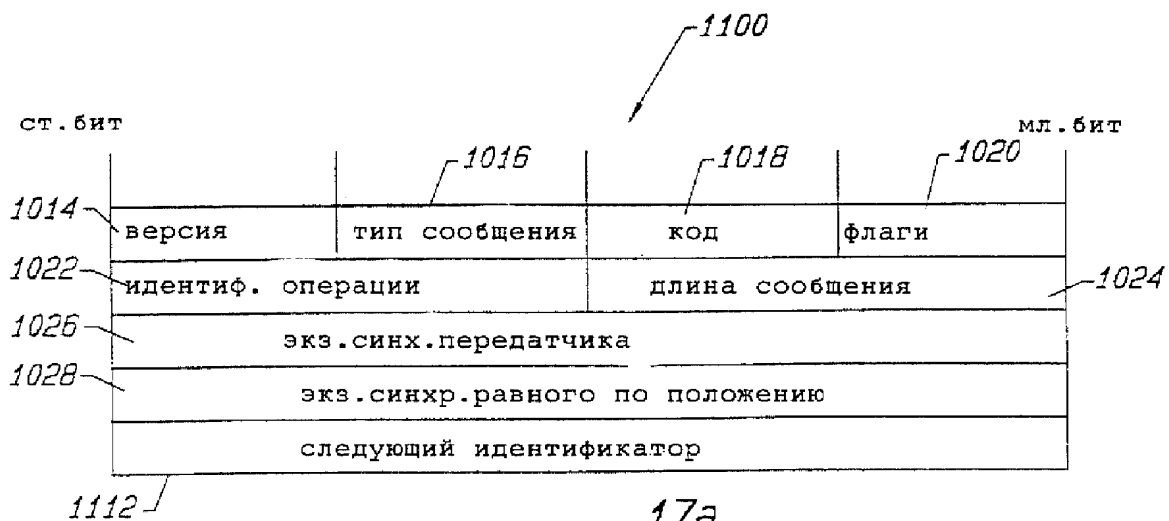
сообщение ответа перечня
интерфейсов протокола



фиг. 17b

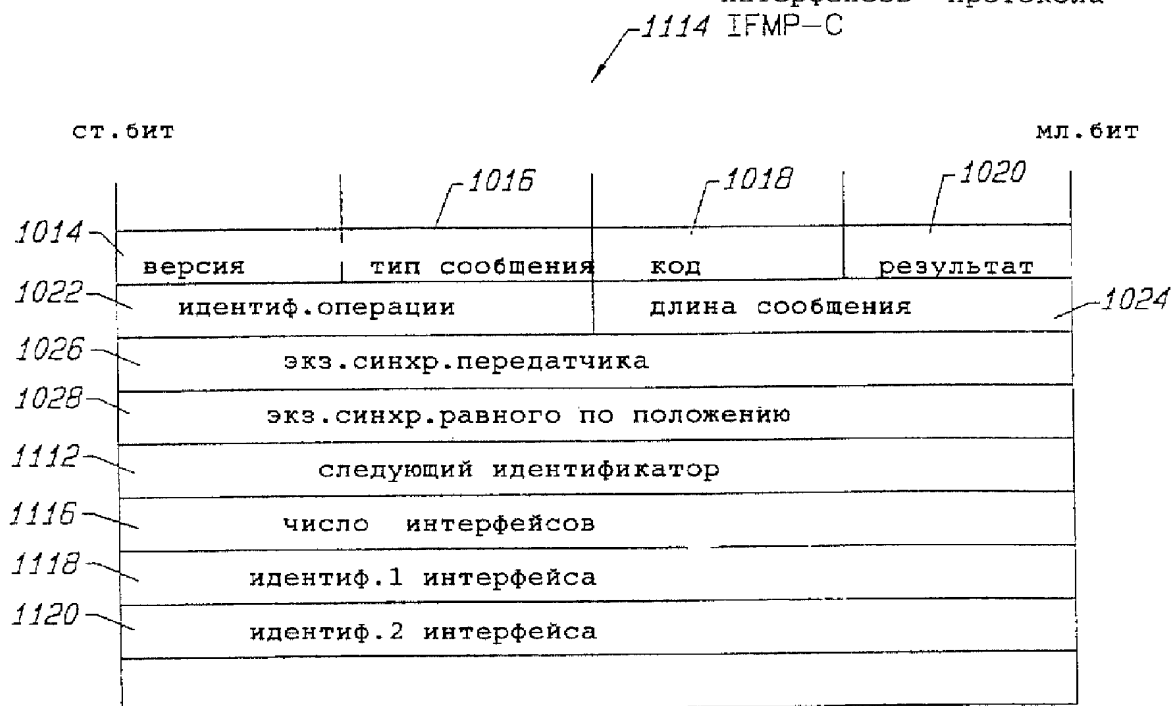
RU 2189072 C2

RU 2189072 C2



Фиг. 17a

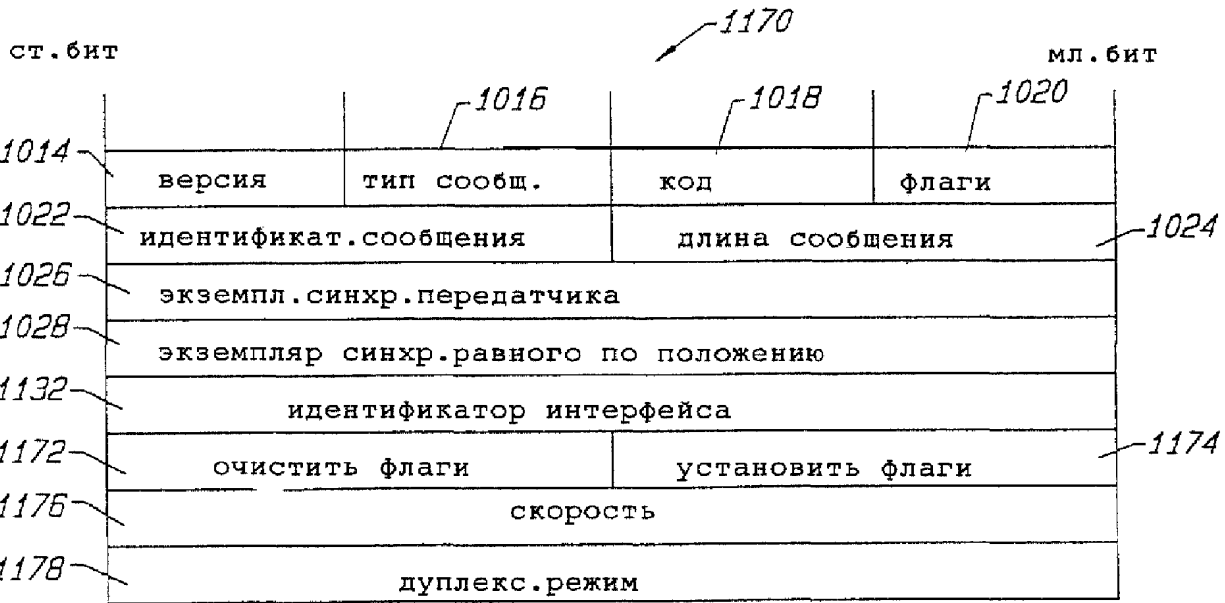
сообщение ответа перечня
интерфейсов протокола



Фиг. 17b

RU 2189072 C2

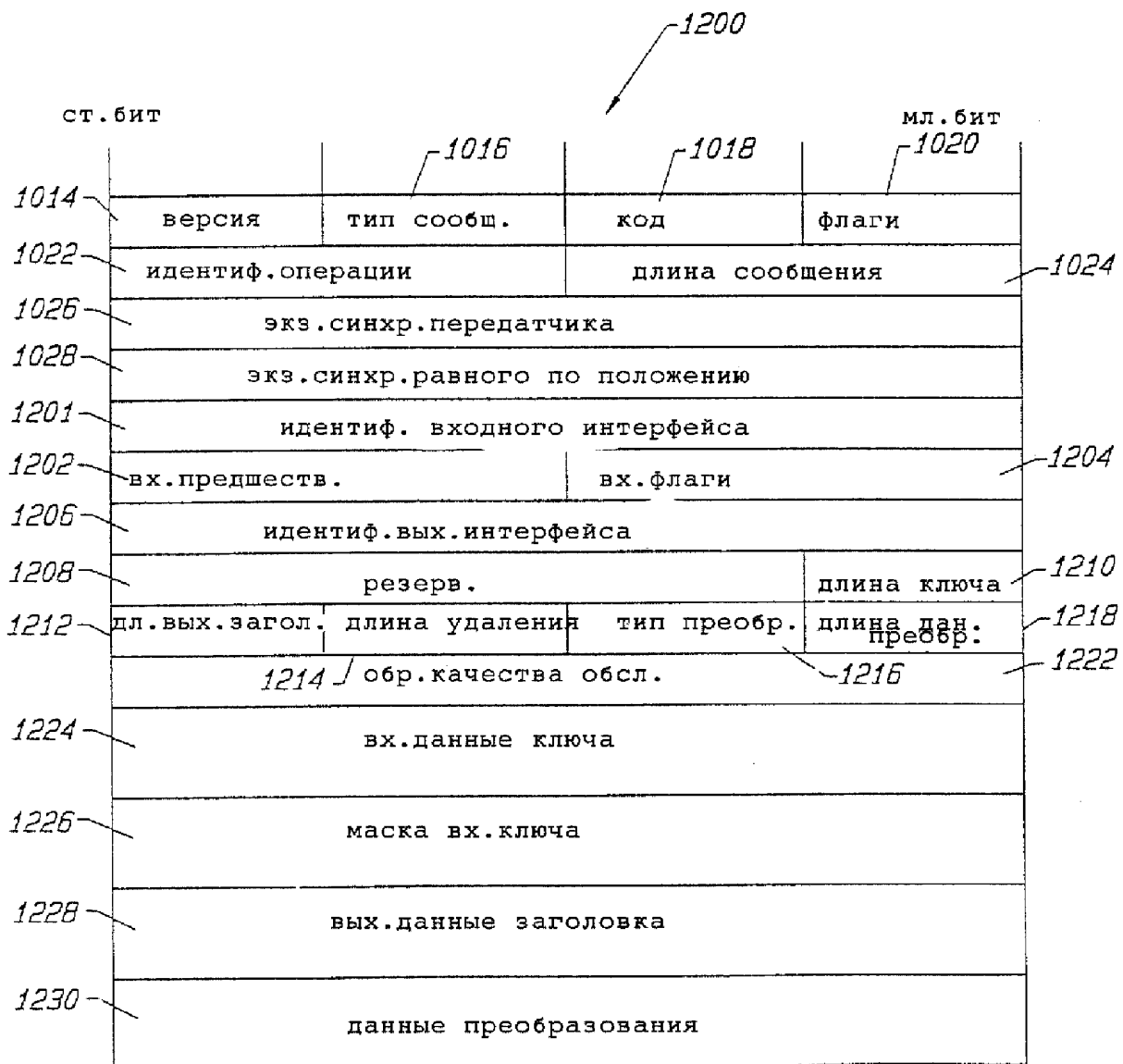
RU 2189072 C2



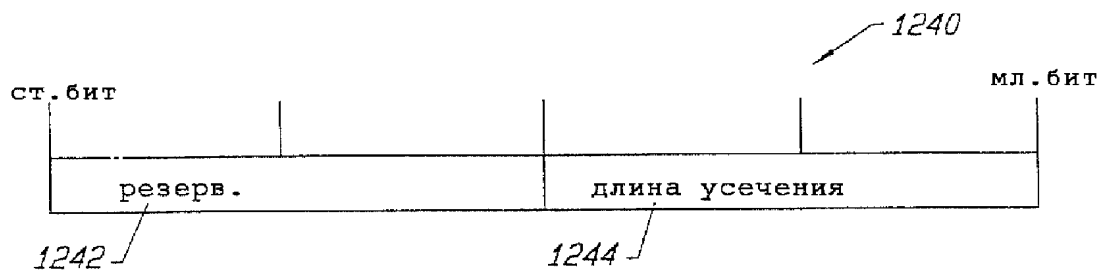
Фиг. 17e

RU 2189072 C2

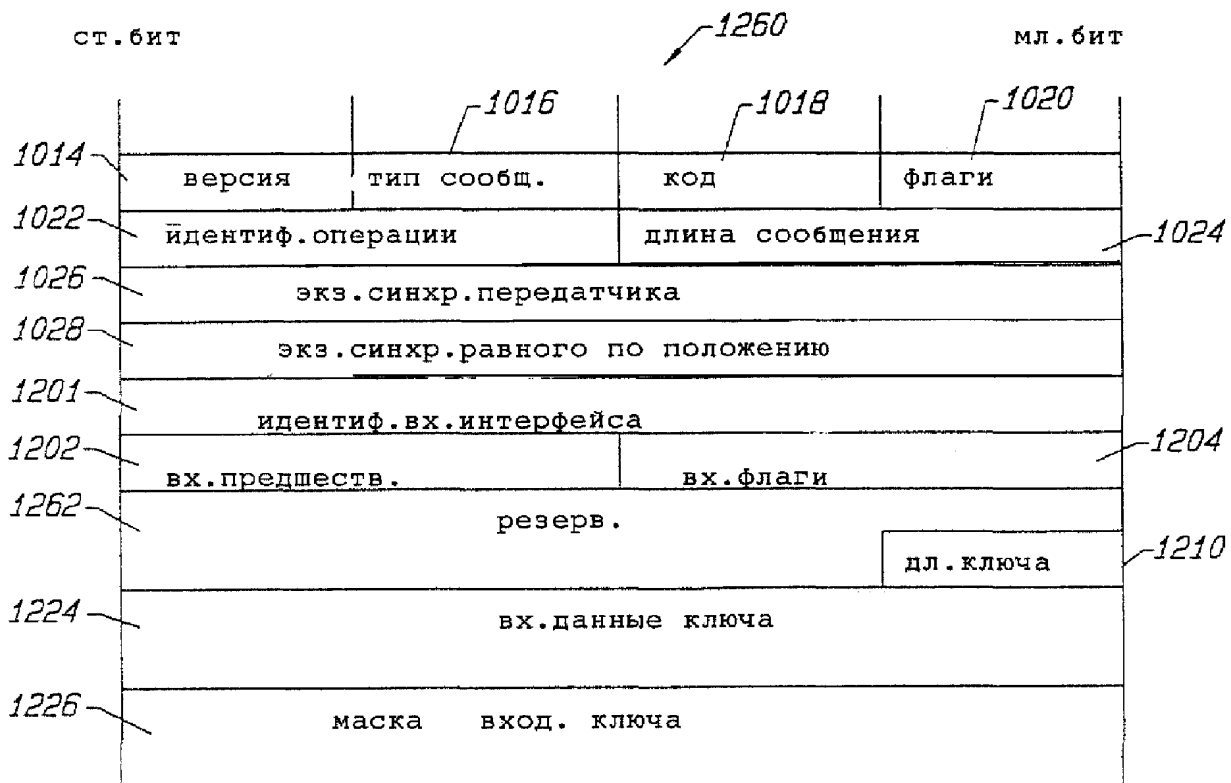
RU 2189072 C2



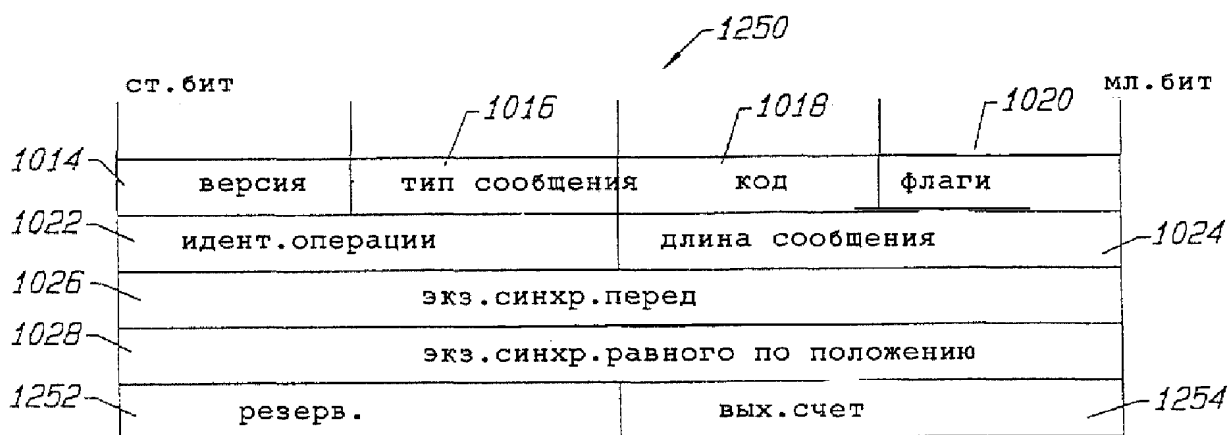
Фиг. 18a



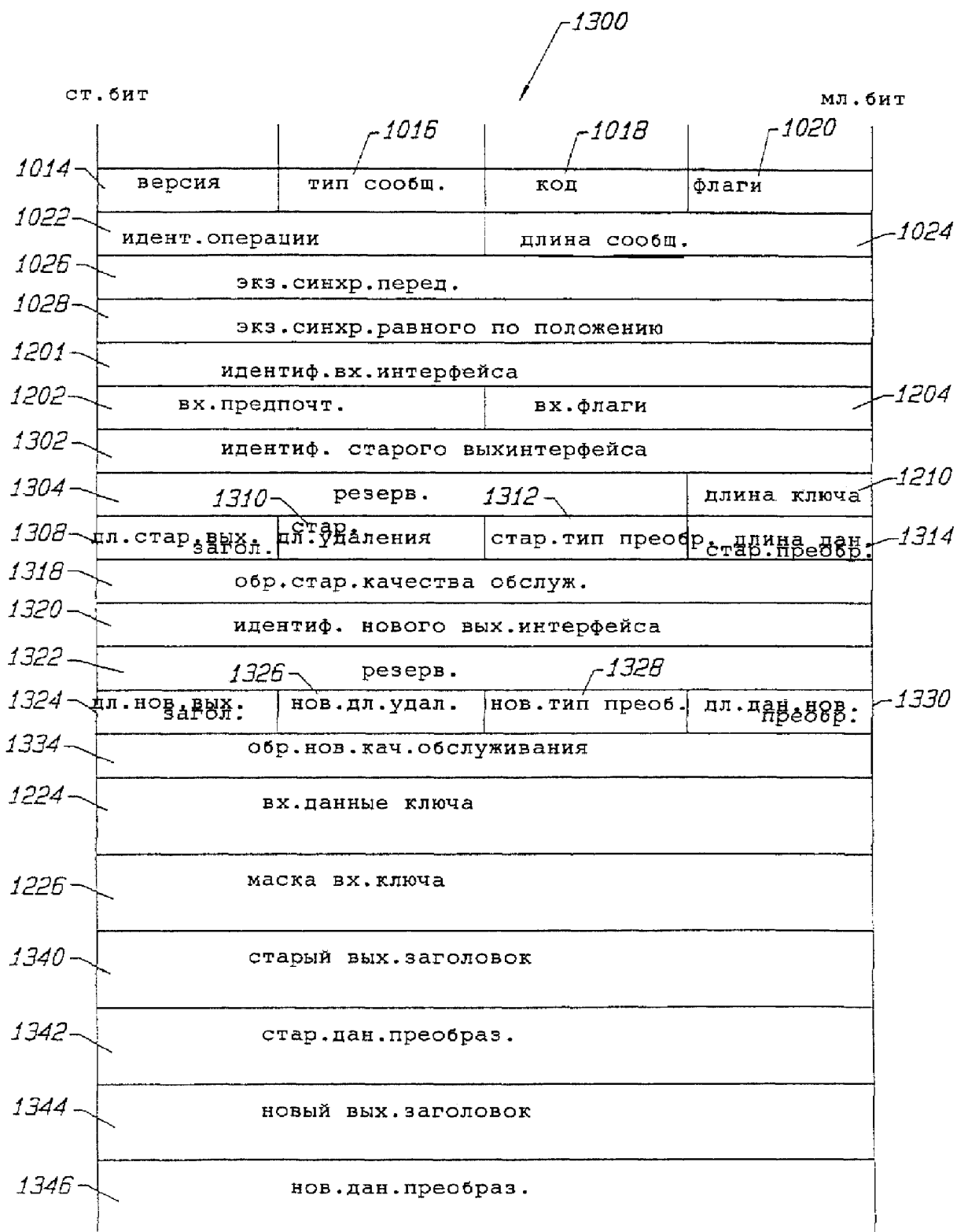
Фиг. 18b



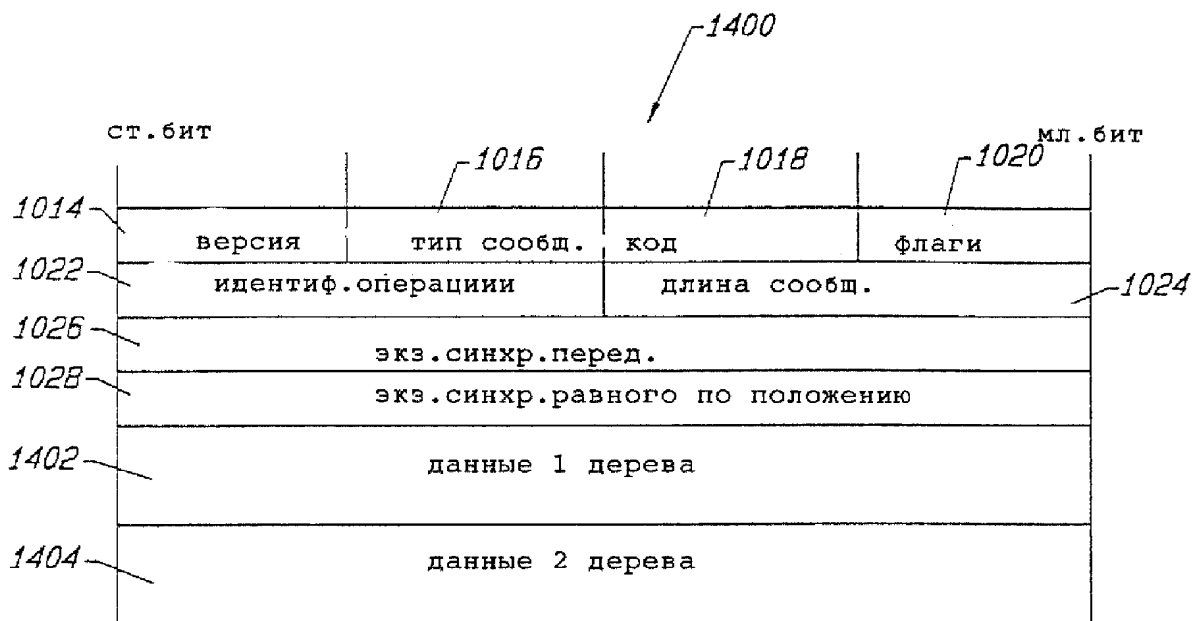
фиг. 18d



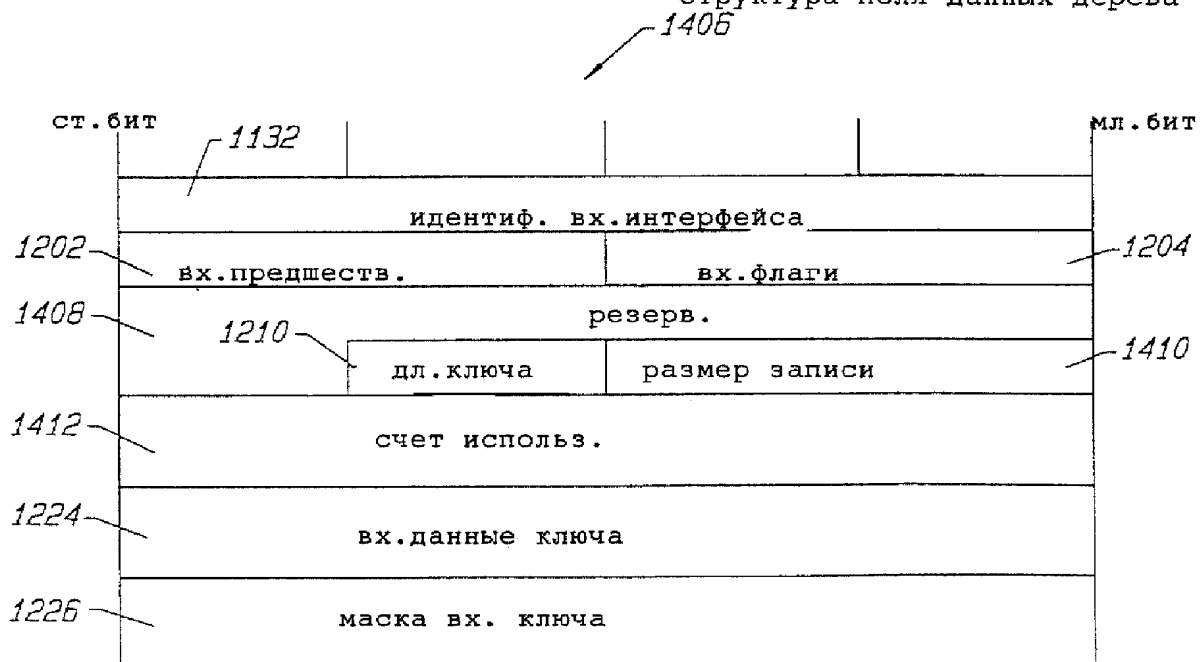
фиг. 18c



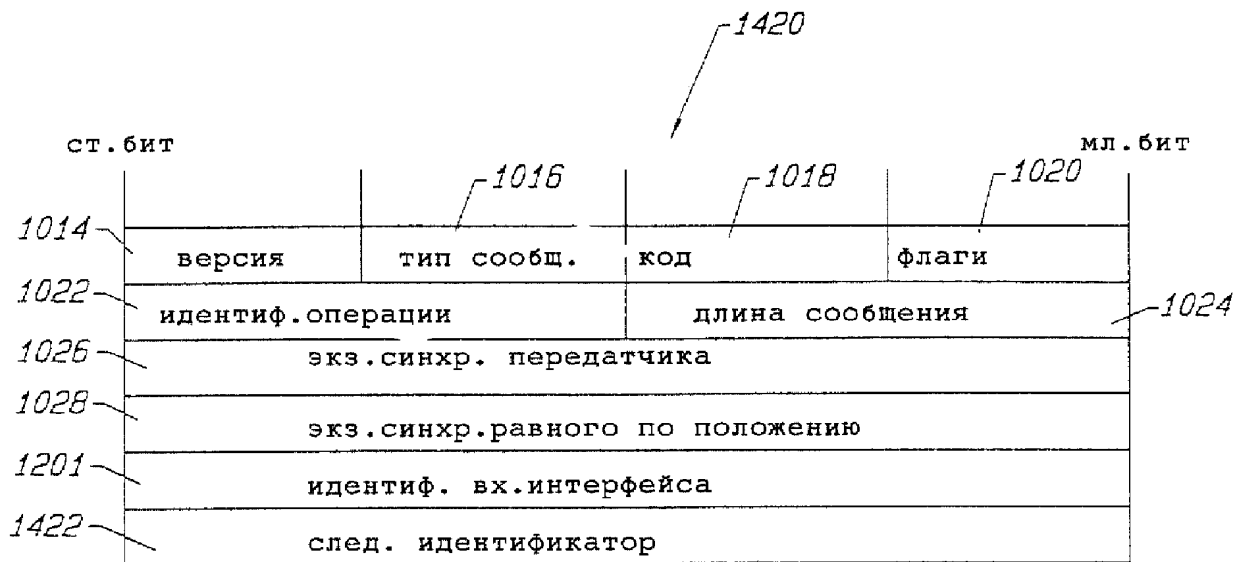
Фиг. 18e



Фиг. 19a
структура поля данных дерева



Фиг. 19b



Фиг. 20a

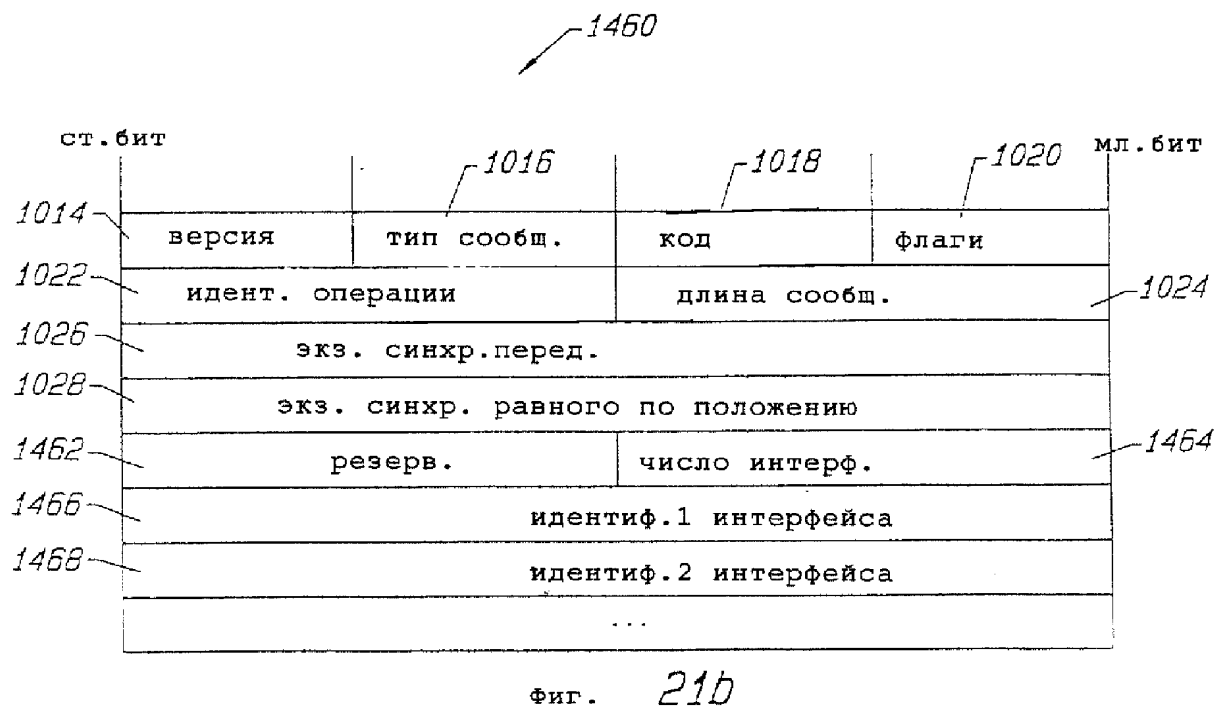
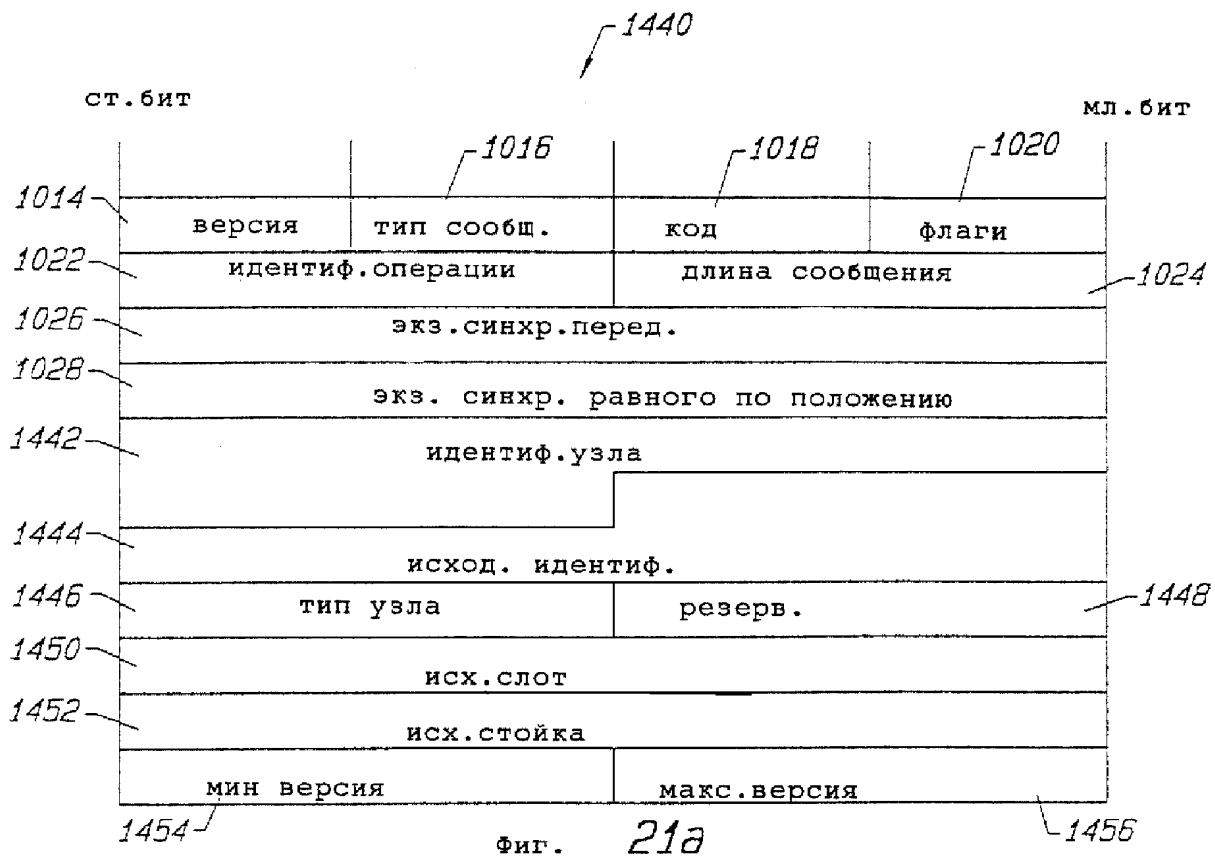
1430

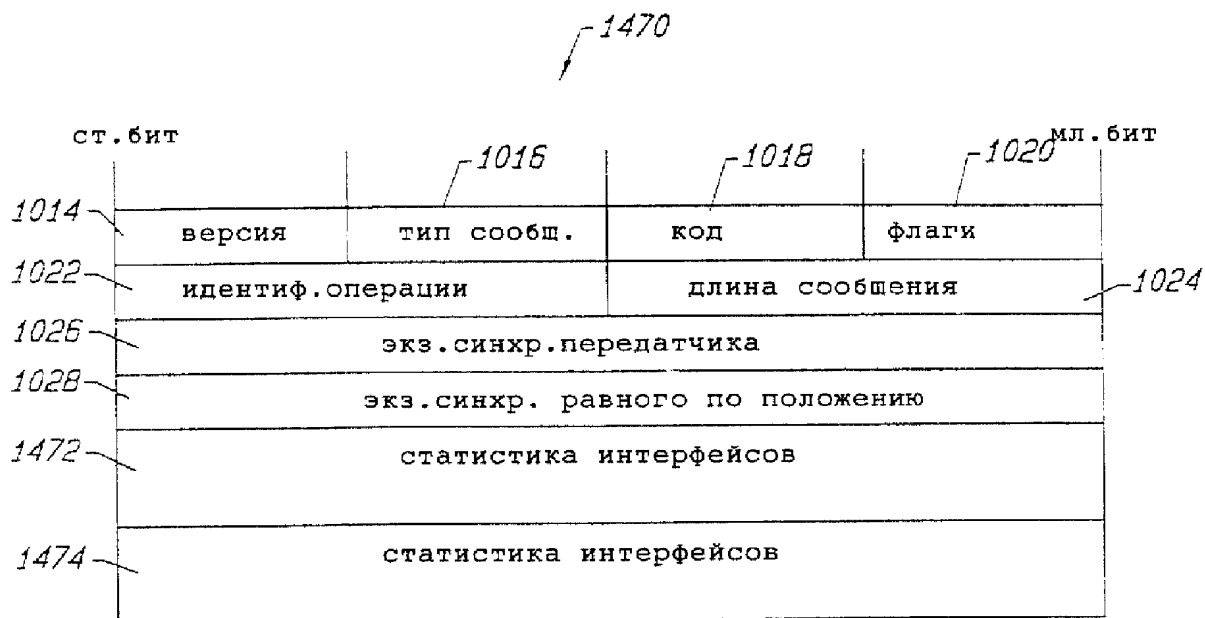
ст. бит		мл. бит		
1014	версия	1016 тип сообщ.	1018 код	1020 флаги
1022	идентиф. операции		длина сообщения 1024	
1026	экз. синхр. перед.			
1028	экз. синхр. равного по положению			
1201	идент. вх. интерфейса			
1202	вх. предшеств.		вх. флаги 1204	
1206	идент. вых. интерфейса			
1432	резерв.			длина ключа
1212	дл. вых. загол.	длина удал.	тип преобр.	длина дан. преобр. 1218
	1214	обр. качества обсл.		1216 1222
1422	след. идентификатор			
1224	вход. данные ключа			
1226	маска вх. ключа			
1228	вых. данные загол.			
1230	данные преобразования			

Фиг. 20b

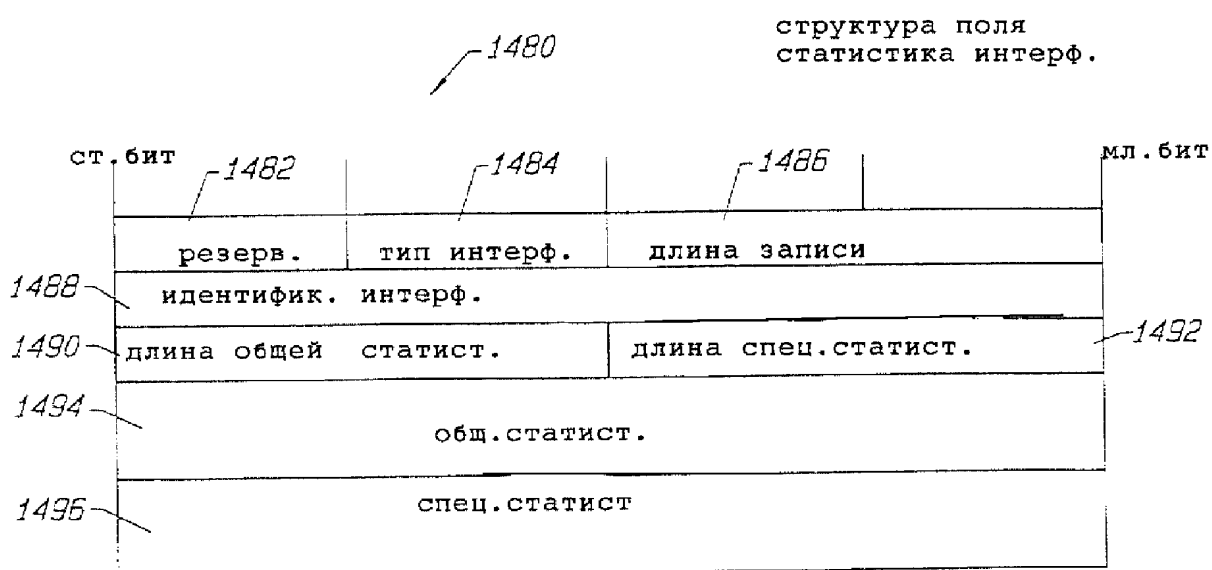
RU 2189072 C2

RU 2189072 C2





Фиг. 21c



Фиг. 21d

1494

структура поля
общстатист.

ст. бит	мл. бит
1500	приним. байты
1502	принятые пакеты групп.передачи
1504	принятые пакеты широковещ.перед.
1506	принят. пакеты однократ.перед.
1508	принят. отказы
1510	принятые ошибки
1512	принят. неизв.
1514	передав. байты
1516	перед.пакеты групп.передачи
1518	перед.пакеты широковещ.передачи
1520	перед.пакеты однократ.передачи
1522	перед. отказы
1524	перед.ошибки

Фиг. 21e

1530 ATM- структура полей спец.статист

ст. бит

1532	принятые элементы данных
1534	переданные элементы данных
1535	ошибки ЦИК
1536	физич. ошибки

Фиг. 21f структура полей спец.статист
тики для Ethernet -данных

1540

ст. бит

1542	принятые ошибки ЦИК
1544	конфликты при передаче

Фиг. 21g

RU 2189072 C2

RU 2189072 C2